**Formation Protocole TCP-IP V4 & netbios et Systèmes Windows – sr41-(sr43-sr22-sr24) - Cours**

Michel Cabaré / www.cabare.net / michel@cabare.net

TCP-IP V4 & netbios et Systèmes Windows  
 - sr41- (sr43- sr 22- sr 24) - Cours V1-7 – Aout 2022

|  |  |  |
| --- | --- | --- |
| https://WWW.CABARE.NET © |  |  |

|  |  |
| --- | --- |
|  |  |

**table des matiÈres**

Structure de TCP/ip 6

Modèle TCP/IP : 6

Couche 1 Interface Réseau : 6

Couche 2 Internet : 6

Couche 3 Transport : 7

Couche 4 Application : 7

Les protocoles de TCP/ip 8

TCP (Transmission Control Protocol) : 8

Port et Socket : 8

Communication en mode Connecté : 8

Fenêtres variables : 9

UDP (User Datagram protocol) : 10

Port et Socket : 10

Communication en mode non Connecté : 10

exemple SNMP 10

exemple Vidéo et Son en ligne 10

IP (Internet Protocol) : 10

Adresse IP 11

Datagramme 11

Fragmentation MTU 12

Assemblage 13

Routage 13

Durée de Vie TTL 13

ICMP (Internet Control Message Protocol) : 14

ARP (Address Resolution Protocol) : 14

Exemple de fonctionnement de ARP en local 14

Exemple de fonctionnement de ARP et Routeur 15

Adresse IP 16

Adresse IP : 16

ID réseau et ID hôte : 17

Classes d'Adresse : 17

Adresses IP Privées : 18

masque de sous-réseau 20

Subdivision de réseau : 20

Masque de sous-réseau : 20

Masque par défaut : 20

Masque personnalisé : 21

Définir un masque de sous-réseau 21

Tables de définition des sous-réseaux : 24

Exemple 6 sous réseaux de 30 postes : 25

masque de sur-réseau 26

Objectif du sur-réseau : 26

Principe : 26

le Routage TCP/IP 28

Notion de routeur : 28

Routage de base : 29

Routage complexe : 30

Table de Routage : 31

Routage statique : 31

Routage dynamique : 31

Réseau Windows 10 32

Gestion Carte Réseau: 32

Désactivation Media Sense: 34

Accès au Centre Réseau et partage : 35

Désactivation Carte Excédentaire : 36

Protocoles LLDP - multiplexage - Topologie réseau Windows: 37

Protocoles Ip-v4 Ip-v6 QoS Client et partage Réseaux 38

Ré-initialiser TCP/IP Sous Windows 10 : 39

Profil – Type Réseau Windows 10 : 41

Changer de type de Profil réseau – interface Paramètre : 42

Changer de type de Profil réseau – Powershell : 43

Changer de type de Profil réseau WI FI – windows 1709 43

Changer de type de Profil réseau – Regedit : 44

Reset - Listes des réseaux identifiés 45

Réseau Windows 7 46

Paramétrage TCP/IP Windows: 46

Accès au Centre Réseau Windows: 47

Profil – Type Réseau seven 7: 48

Choisir un profil Réseau 7 : 48

Profil réseau avancé – Voisinage Réseau 51

Réglage Disponibles: 51

Jeux de Réglages: 52

Activer la découverte du "voisinage réseau": 53

Mécanisme du Voisinage réseau 55

Principe de fonctionnement : 55

Rafraîchissement Tests et vérifications : 56

Peut on éviter l'élection d'un Explorateur ? : 56

Protocole DHCP 59

Objectif de DHCP : 59

Fonctionnement de DHCP : 59

DHCPDISCOVER ou "Demande de bail IP" : 60

DHCPOFFER ou "Offre de bail IP" : 60

DHCPREQUEST ou "Selection de bail IP" : 60

DHCPACK / NACK ou "Accusé de réception de bail IP" : 60

"Renouvellement de bail IP" : 61

DHCPRELEASE ou libération des ressources: 61

Client DHCP 62

Client DHCP Windows 10 - Seven 62

Ipconfig /release /renew : 63

Adresses Ip automatiques (APIPA) 64

Principe APIPA et DHCP: 64

APIPA et Windows: 64

Désactivation adresse APIPA: 64

Adresse IP alternative: 65

Notion de DNS 66

Le DNS: 66

Noms DNS 66

Nom "Plat" Netbios 66

Nom "Hierarchique" DNS 66

Structure des domaines – délégation de zones 67

Zones DNS: 68

Zone principale – secondaire 69

Requêtes itératives ou récursives 69

Résolution de Noms et Résolution inverse 70

Ordre de Résolution DNS par le client Windows : 70

Nom netbios 72

Protocole NetBeui : 72

Résolution de nom NetBIOS 73

Paramétrer la résolution NetBIOS 74

Nom Netbios - Nom d'hôte: 75

Intérpretation des Nom NetBios : 76

Hosts - LMhosts 79

Fichier Hosts et LMHosts: 79

Fichier lmHosts (nom netbios): 79

Détails écriture lmhosts 80

Fichier hosts (nom d'hôte): 80

annexe : Trames tcp/IP 81

Broadcast : 81

Unicast : 82

Multicast : 83

Dossier ..\System32\driver\etc 84

Fichiers exemples Windows : 84

TP - WorkGroup entre réseaux 85

1 réseau IP et x Workgroups différents: 85

test et vérification : 85

TP - Modifier lmhosts 86

Inscrire une machine simple dans lmhosts : 86

Inscrire un Contrôleur de Domaine dans lmhosts : 86

TP - Modifier hosts 88

Inscrire une machine dans hosts : 88

Interdire une machine un site dans hosts : 89

Tester TCP/IP 90

ICMP et l'Utilitaire PING : 90

Types de réponses à un ping 90

Méthodologie de test 91

Ping –a 92

Ping –t 92

Test TTL ping -i: 92

Tracert : 94

Pathping : 94

Ipconfig.exe /all: 95

ARP et l'Utilitaire ARP –a : 96

Arp –a 96

Usurpation d’adresse ARP : 97

Getmac /v: 98

A distance = ping + arp / nbtstat -A: 98

Test de DNS 99

Test DNS d’un client d’un domaine : 99

Nom d’hôte et FQDN 99

Nslookup en mode interactif 99

Nslookup et non-réponse de Serveur Windows : 103

Nslookup et Ping : 104

Serveur DNS public – connus : 104

Tester TCp-IP - netstat 105

Netstat: 105

Netstat -a n port en écoute: 106

Netstat -a –p TCP port en écoute par protocole: 106

Test liaison ftp – affichage dans Netstat –an : 107

Nbtstat –n : 109

Telnet test de socket 110

Installation telnet: 110

Telnet - Test de Socket = @IP+ port distant: 111

Port 3389 (RDP) 111

Port 22 (SFTP) 112

Port 21 (FTP) 112

Tester TCP/IP - Compléments 113

Test MTU ping -l -f: 113

Constat de la valeur MTU 1500 en Wan 113

Jumbo Frames - MTU en Lan 115

Routage route print netstat –r : 116

Test routage route add : 117

# Structure de TCP/ip

## Modèle TCP/IP :

Par rapport au modèle OSI classique en 7 couches, le modèle présentant **TCP/IP** est composé de 4 couches uniquement :

|  |  |
| --- | --- |
| **OSI** | **TCP/IP** |
| ➐Application  ➏Présentation  ➎Session | ➍Application :**SNMP-FTP-SMTP**… |
| ➍Transport | ➌Transport : **TCP** ou **UDP** |
| ➌Réseau (routage) | ➋Internet : **IP, ARP, ICMP** routage : **RIP, SPF** |
| ➋Liaison  ➊Physique | ➊Interface Réseau |

## Couche 1 Interface Réseau :

Elle a en charge la communication physique avec le réseau. Par conséquent doit pouvoir accepter les normes **Ethernet**, **Token-Ring**…

## Couche 2 Internet :

Elle s'occupe du routage et de la livraison des paquets au travers du protocole **IP (Internet protocol)**.

Tous les protocoles de la couche Transport passent par **IP** pour acheminer leurs données, mais IP est un protocole non connecté, il ne garantit pas donc que les paquets émis ne soient pas perdus, dupliqués ou inutilisables…

C'est aux couches supérieures (transport ou application) de vérifier le résultat!

La couche internet contient aussi un **protocole ICMP (Internet Control Messaging Protocol)** permettant de mettre en œuvre des contrôles sur le transport des paquets IP et de rapporter les erreurs…

La couche internet contient aussi un protocole **ARP (Adress resolution Protocol)** permettant de mettre en œuvre des mécanismes de résolution pour trouver une adresse physique avec une adresse IP…

## Couche 3 Transport :

Elle a elle le rôle de fournir à la couche application une communication entre 2 machines…

2 protocoles existent selon que l'on souhaite utiliser une communication avec connexion ou sans…

le protocole **TCP (transmission Control Protocol)** est utilisé pour la communication connectée entre deux machines (fiable mais avec un débit relativement faible du fait des contrôles)

le protocole **UDP (user Datagram Protocol)** est utilisé pour la communication non connectée, sans garantie de distribution (moins fiable mais avec un débit plus élevé du fait de l'absence des vérifications)

## Couche 4 Application :

Elle prend en charge toutes les activités supérieures du modèle OSI.

Plusieurs protocoles existent dans cette couche selon l'objectif visé :

**SNMP (Simple Network management Protocol)** -> gestion de réseau

**FTP (File transfer protocol)** -> transfert de fichier

**SMTP (Simple Mail transfer Protocol)** -> courrier électronique

**HTTP (Hyper Text Transfer Protocol)** -> serveurs web

# Les protocoles de TCP/ip

## TCP (Transmission Control Protocol) :

**TCP**, on l'a dit, est un protocole utilisé pour la communication connectée entre deux machines (fiable mais avec un débit relativement faible du fait des contrôles)

### Port et Socket :

Les **Ports** identifient les processus en cours d'exécution dans la couche application, et par conséquent un n° de port identifie un processus auquel on doit envoyer des données.

Les numéros de ports sont donnés de manière prédéterminée pour ceux allant de 1 à 1023, mais restent libres pour les autres

Il a été ainsi arbitrairement décidé d'un N° de Port pour chaque usage.

Port n° 21 : File Transfer Protocol

Port n° 22 : SSL connexion à distance sécurisée

Port n° 23 : Telnet

Port n° 25 : SMTP réception de courrier

Port n° 53 : DNS Domain Name Server

Port n° 80 : HTTP pages web

Port n° 88 : Kerberos authentification (NT 2000)

Port n° 110 : POP3 lecture de courrier

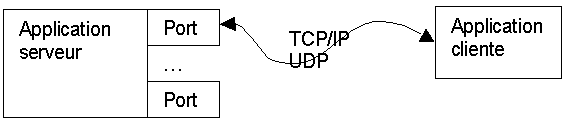
Port n° 137 à 139 : NetBios

Port n° 443 : HTTPS pages web sécurisées

Port n° 546 : DHCP

Les ports proposent 65535 point d'accès à un ordinateur à partir d'une seule adresse physique.

L'ensemble d'une adresse IP d'un ordinateur essayant de communiquer et du numéro de ports utilisé crée ce que l'on appelle un "**Socket**"



### Communication en mode Connecté :

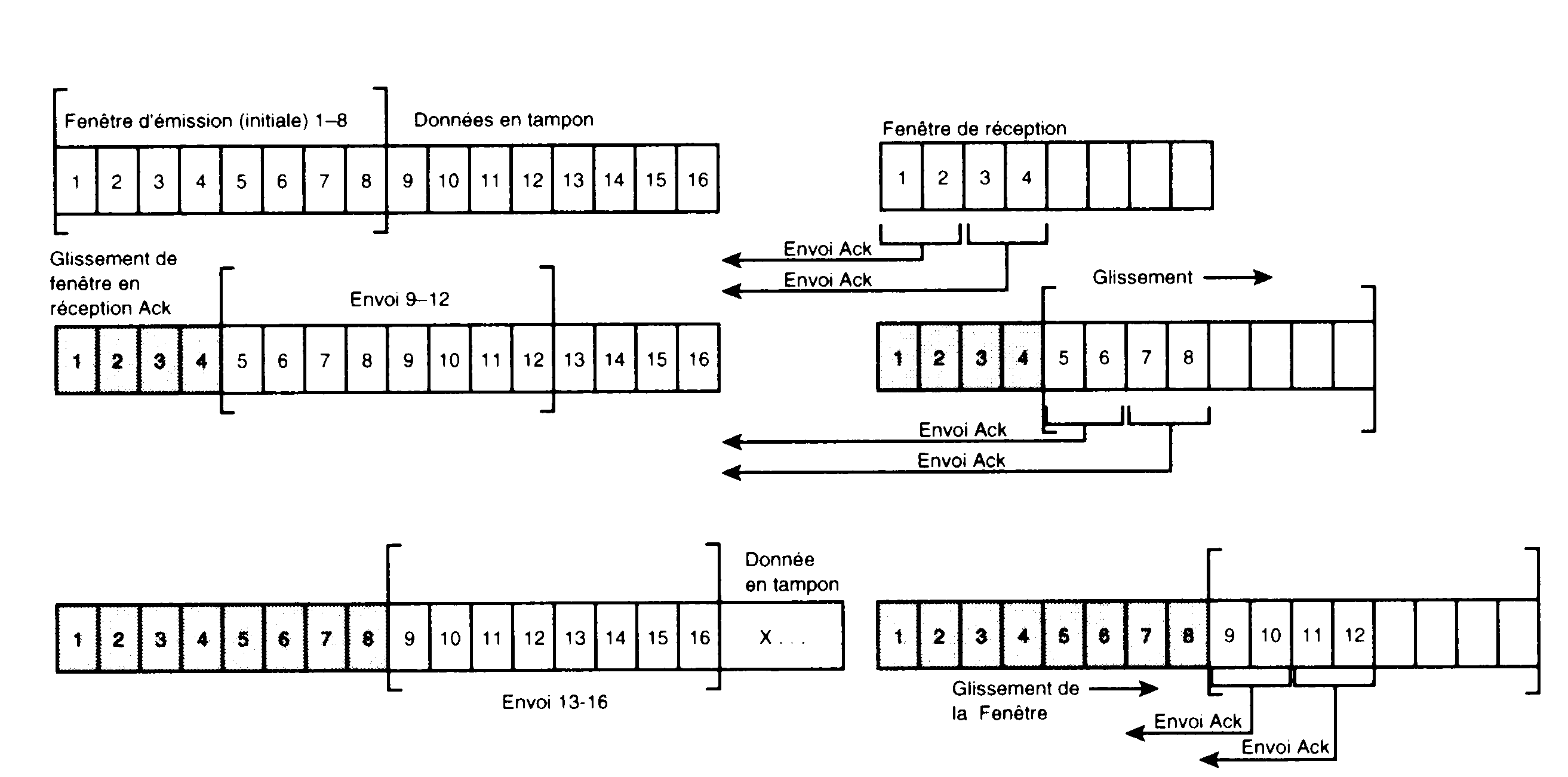
TCP demande qu'une session soit établie avant de transmettre les données entre les machines connectées.

En tant que protocole en mode connecté, TCP suit la transmission et la réception des paquets individuels durant la communication. TCP envoie les paquets en séquences et demande un accusé de réception de ces paquets avant d'en envoyer d'autres.

Etant donné le mécanisme de vérification effectué par TCP, le format d'un paquet TCP peut être assez complexe…

### Fenêtres variables :

Chaque machine dispose d'une **fenêtre d'émission** et **d'une fenêtre de réception** qu'elle utilise comme tampon de donnée pour rendre la communication plus efficace.



Une **fenêtre de réception** permet à une machine de recevoir des paquets en désordre (en effet TCP utilise IP qui ne garantit pas l'ordre d'arrivée, ni même l'arrivée des paquets !) et de les classer pendant qu'elle attend les paquets suivants

Au fur et a mesure que la fenêtre de réception récupère des paquets, elle renvoi des accusés de réception (un accusé tous les 2 paquets reçuts)

Si la fenêtre d'émission ne reçoit pas d'accusé de réception, elle attends puis retransmet les paquet non acquittés .

**N.B** : Dans la **fenêtre d'émission** un temporisateur est positionné pour chaque paquet envoyé, indiquant le temps à attendre avant d'estimer que le paquet n'est pas arrivé. En cas de non acquittement, le paquet est envoyé une nouvelle fois avec le temporisateur doublé, après cette nouvelle attente, s'il n'y a toujours pas d'acquittement, on recommence en doublant encore le temporisateur…avec un maximum de x tentatives…

**N.B:** Sous WINDOWS les fenêtres par défaut ont une taille de 8 kilo-octets, soit 8 trames Ethernet standard

**N.B:** Sous WINDOWS les fenêtres d'émission sont paramétrées par défaut pour tenter d'émettre 5 fois maximum

**N.B:** Lors d’une latence faible, la vitesse d’émission augmente progressivement, mais avec une latence forte, la vitesse d’émission va chuter !

## UDP (User Datagram protocol) :

**UDP**, on l'a dit, est utilisé pour la communication non connectée, sans garantie de distribution (moins fiable mais avec un débit plus élevé du fait de l'absence des vérifications)

### Port et Socket :

Les paquets **UDP** sont transmis comme pour TCP a des **Sockets**, c'est à dire à des couples adresses Ip + N° de Port, mais avec moins de fiabilité (puisque aucun contrôle n'est effectué…)

Port n° 67-68 : SNMP gestion - surveillance réseau

Port n° 520 : RIP routage IP dynamique

### Communication en mode non Connecté :

On peut se demander où réside l'intérêt d'un tel protocole, fondamentalement dans sa faible surcharge (les données qu'il rajoute pour sa gestion sont très faibles par rapport aux données utiles transmises…)

Deux exemples suffiront à se convaincre de l'intérêt de ce protocole

### exemple SNMP

SNMP utilise le protocole **UDP** pour véhiculer ses interrogations sur le réseau, et transmettre les messages d'erreurs d'une machine…

Il est normal que lorsque une machine soit défaillante, elle ne puisse réussir à mettre en place une session **TCP** pour transmettre son … malaise !

une diffusion **UDP** est beaucoup plus raisonnable ne terme "espérance de vie" de la part de cette machine

### exemple Vidéo et Son en ligne

Dans ce cas de figure il faut privilégier à tout prix le débit, ce que **UDP** fait, au détriment du paquet perdu, qu'il est bon d'ailleurs de ne pas tenter de réémettre…

En effet si on écoute un morceau de musique, et qu'un segment manque, notre oreille s'en rend à peine compte, et notre "cerveau" corrige ! Imaginons l'effet auditif du lecteur de CD qui bloque l'émission pour attendre la réception acquittée du fragment retardataire…

## IP (Internet Protocol) :

Tous les protocoles de la couche Transport passent par **IP** pour acheminer leurs données, mais IP est un protocole non connecté, il ne garantit pas donc que les paquets émis ne soient pas perdus, dupliqués ou inutilisables…

C'est au couches supérieures (soit via TCP dans le transport ou application si on utilise UDP dans le transport) de vérifier le résultat!

### Adresse IP

Ce protocole repose en partie sur la notion d'adresse IP (Internet Protocol) décernée de façon unique pour chaque élément matériel faisant partie d'un réseau

on verra cette notion en détail dans le chapitre "Adresse IP" (page 16 )

### Datagramme

**IP** reçoit des information des protocoles **TCP** ou **UDP** et les renvoi dans ce que l'on appelle un **Datagramme**, c'est à dire un bloc de donnée dans lequel IP à rajouté ses informations (type de protocole utilisé : udp ou tcp, adresse ip de la machine d'origine, adresse ip de la machine destinataire, durée de vie…) aux données utiles.

Les données qui circulent sur Internet sous forme de datagrammes (on parle aussi de paquets) sont des données encapsulées, c'est-à-dire des données auxquelles on a ajouté des en-têtes correspondant à des informations sur leur transport (telles que l'adresse IP de destination, ...).

Les données contenues dans les datagrammes sont analysées (et éventuellement modifiées) par les routeurs permettant leur transit.

Voici ce à quoi ressemble un datagramme:

<----------------------------- 32 bits ------------------------------>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Taille d'en-tête | type de service | Longueur totale | |
| Identification | | | Drapeau | Décalage fragment |
| Durée de vie | | Protocole | Somme de contrôle en-tête | |
| Adresse IP source | | | | |
| Adresse IP destination | | | | |
| Données | | | | |

Voici la signification des différents champs:

* + - **Version**: il s'agit de la version du protocole IP que l'on utilise (actuellement on utilise la version 4 *IPv4*) afin de vérifier la validité du datagramme. Elle est codée sur 4 bits
    - **Taille d'en-tête**: il s'agit du nombre de mots de 32 bits sur lesquels sont répartis l'en-tête
    - **Type de service**: il indique la façon de laquelle le datagramme doit être traité
    - **Longueur totale**: il indique la taille totale du datagramme en octets. La taille de ce champ étant de 2 octets, la taille totale du datagramme ne peut dépasser 65536 octets. Utilisé conjointement avec la taille de l'en-tête, ce champ permet de déterminer où sont situées les données
    - **Identification, drapeaux (flags) et déplacement de fragment** sont des champs qui permettent la fragmentation des datagrammes, il sont expliqués plus loindans l'assemblage**.**
    - **Durée de vie: (appelée aussi TTL: Time To Live)** indique le nombre maximal de routeurs à travers lesquels le datagramme peut passer. Ainsi ce champ est décrémenté à chaque passage dans un routeur, lorsque celui-ci atteint la valeur critique de 0, le routeur détruit le datagramme. Cela évite l'encombrement du réseau par les datagrammes perdus
    - **Protocole:** ce champ permet de savoir de quel protocole est issu le datagramme avec par exemple   
       ICMP: 1   
       IGMP: 2   
       TCP: 6   
       UDP: 17
    - **Somme de contrôle de l'en-tête (header checksum):** ce champ contient une valeur codée sur 16 bits qui permet de contrôler l'intégrité de l'en-tête afin de déterminé si celui-ci n'a pas été altéré pendant la transmission.
    - **Adresse IP Source:** Ce champ représente l'adresse IP de la machine émettrice, il permet au destinataire de répondre
    - **Adresse IP destination:** Adresse IP du destinataire du message

### Fragmentation MTU

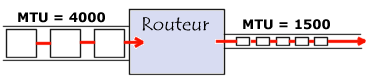
La taille d'un Datagramme dépendant du type de réseau utilisé, Ethernet, Token-Ring…IP doit alors éventuellement découper les données qu'il reçoit de TCP ou de UDP en morceau pour être émises dans plusieurs Datagrammes de taille adéquate. Ce découpage, avec repérage et étiquetage des morceaux s'appelle la Fragmentation.

la taille d'un datagramme maximale est de 65535 octets. Toutefois cette valeur n'est jamais atteinte car les réseaux n'ont pas une capacité suffisante pour envoyer de si gros paquets. De plus, les réseaux sur Internet utilisent différentes technologies, si bien que la taille maximale d'un datagramme varie suivant le type de réseau.

La taille maximale d'une trame est appelée ***MTU*** (**Maximum Transfer Unit**), elle entraînera la fragmentation du datagramme si celui-ci a une taille plus importante que le MTU du réseau.

|  |  |
| --- | --- |
| Type de réseau | MTU (en octets) |
| Arpanet | 1000 |
| Ethernet | 1500 |
| FDDI | 4470 |

La fragmentation d'un datagramme se fait au niveau des routeurs, c'est-à-dire lors de la transition d'un réseau dont les MTU sont différents



### Assemblage

Bien sûr à l'arrivée **IP** doit récupérer tous les morceaux et reconstruire les données d'origine, cela s'appelle l'**Assemblage**.

Le routeur va donc ensuite envoyer ces fragments de manière indépendante et ré-encapsulé (il ajoute un en-tête à chaque fragment) de telle façon à tenir compte de la nouvelle taille du fragment, et en ajoutant des informations afin que la machine de destination puisse réassembler les fragments dans le bon ordre (rien ne dit que les fragments vont arriver dans le bon ordre étant donné qu'ils sont acheminés indépendamment les uns des autres...).

Un datagramme possède plusieurs champs pour calculer l'assemblage:

### Routage

Le protocole **IP** doit **router** les datagrammes d'un réseau à l'autre. Toutes les machines d'un réseau ne sont pas des routeurs, mais un **routeur** est une machine qui lorsqu'elle reçoit un datagramme qui ne lui est pas adressé, doit renvoyer ce paquet sur le réseau dans la bonne direction pour qu'il atteigne sa destination…Le principe du routage IP peut être résumé ainsi:

1. **Extraire** l'adresse IP de destination du datagramme.
2. Appliquer à cette adresse le masque de sous-réseau éventuel  
   (ET logique entre l'adresse et le masque).
3. Extraire la partie "**Identificateur Réseau**" de l'adresse ainsi obtenue.
4. S'agit-il du réseau local ?
   * Si oui, procéder à l'encapsulation et au **routage direct**.
   * Sinon, existe-t-il une entrée dans la **table de routage** pour ce réseau de destination ?
     1. Si oui, envoyer le datagramme vers la passerelle spécifiée dans la table.
     2. Sinon, existe-t-il une **route par défaut** ?
        + Si oui, envoyer le datagramme à la passerelle spécifiée par la route par défaut.
        + Sinon, déclarer une **erreur** de routage. (Protocole ICMP)

### Durée de Vie TTL

La durée de vie ou **TTL (Time To Live)** correspond à l'idée suivante. Chaque fois qu'un **Datagramme** prends le départ d'une machine sur le réseau vers une destination connue, il a une espérance de vie exprimée en seconde.

A chaque passage dans un **routeur**, celui-ci décrémente de 1 seconde son compteur **TTL** de vie, de sorte que si le datagramme tarde trop à parvenir à la machine destinataire, un routeur le "détruira" en réduisant son **TTL** à 0

**N.B:** Sous Windows les Datagrammes ont une valeur de Vie par défaut de 128, Sous linux la valeur est à 64

## ICMP (Internet Control Message Protocol) :

**ICMP** permet de mettre en œuvre des contrôles sur le transport des paquet IP et de rapporter les erreurs…

Les messages **ICMP** servent principalement à rapporter des erreurs et envoyer des requêtes.

Dans la pratique on utilise le protocole **ICMP** essentiellement pour envoyer des requêtes **d'Echo request** et pour attendre des réponses **d'Echo reply**, et encore ceci à travers un utilitaire **PING (Personnal Internet Groper)**

On verra cette notion en détail dans le chapitre "Tester IP" (page 90 )

## ARP (Address Resolution Protocol) :

A part dans le cas ou on émet en diffusion, lorsque IP souhaite émettre, il doit connaître l'adresse physique, ou adresse mac, ou adresse Ethernet du poste destinataire

**ARP** permet de mettre en œuvre des mécanismes de résolution pour trouver l'adresse physique correspondant à une adresse IP locale…

Si **ARP** ne connaît pas l'adresse physique de l'adresse IP locale demandée, il fonctionne par diffusion locale et, une fois trouvée, stocke cette correspondance dans sa mémoire ( pendant un certain temps)

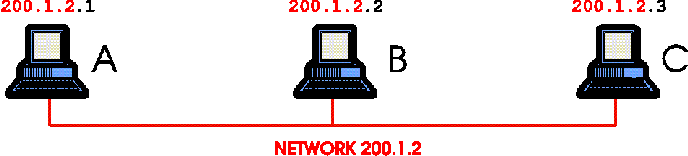
Si **ARP** connaît l'adresse physique dans son cache, il ne diffuse rien sur le réseau et cela fonctionne très bien

**N.B** : Mais **ARP** ne peut trouver que des adresses physiques locales, et il ne retourne jamais à IP une adresse physique qui se trouve sur un réseau distant ! Dans ce Cas **IP** (qui peut via l'adresse ip se rendre compte que la machine demandée n'est pas une machine locale) ne demande pas à **ARP** de trouver l'adresse physique de la machine distante, mais il lui demande de trouver l'adresse physique du **routeur** !

On verra cette notion en détail dans le chapitre "Tester IP" (page 90 )

### Exemple de fonctionnement de ARP en local

Soit un réseau interne TCP/IP comprenant un segment Ethernet et trois machines. Le numéro de réseau IP de ce segment est 200.1.2. Les numéro d'hôte pour A, B et C sont 1, 2 et 3 respectivement. Ce sont des adresses de classe C, ce qui permet d'avoir 254 machines sur ce segment.

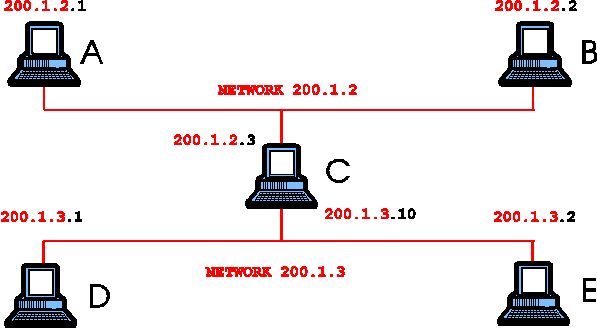


Supposons que A veuille envoyer un paquet à C pour la première fois, et qu'il connait l'adresse IP de C. Pour envoyer ce paquet sur ce brin Ethernet, A aura besoin de connaître l'adresse MAC (ou adresse Ethernet) de C. Le protocole **ARP** (**Address Resolution Protocol**) est utilisé pour trouver dynamiquement cette adresse.

ARP garde une table interne d'adresses IP et d'adresses MAC correspondantes. Quand A essaye d'envoyer un paquet IP à C, le module d'ARP consulte sa table d'adresses IP et ne découvrira aucune entrée pour C. ARP envoie alors un paquet spécial reçu par tous (broadcast), demandant l'adresse MAC correspondant à l'adresse IP qu'il connait. S'il n'y a pas de "time-out", cela signifie que la machine C a répondu en incluant son adresse MAC dans sa réponse, et le tour est joué. A met à jour sa table d'adresse (ou table d'hôte) et peut envoyer son paquet.

### Exemple de fonctionnement de ARP et Routeur

Considérons maintenant 2 réseaux Ethernet séparés et reliés par la machine C, fonctionnant comme un routeur.

La machine C agit comme un routeur entre ces deux réseaux. Un routeur est un élément qui choisit différentes directions pour les paquets en fonction de l’adresse IP. Comme il y a deux segments Ethernet séparés, chaque réseau a son propre numéro de réseau de classe C. Ceci est indispensable car le routeur ne connaît que des interfaces sont associées à un réseau.

Si A veut envoyer un paquet à E, il doit d’abord l’envoyer à C qui peut faire suivre le paquet à E. Ceci est possible car A utilise l’adresse MAC de C et l’adresse IP de E. C va donc recevoir le paquet destiné à E et va le faire suivre en utilisant l’adresse MAC de E, soit parce qu’il la connaît, soit en faisant une requête ARP comme décrit précédemment.

Si E reçoit le même numéro de réseau que A, soit "200.1.2", A essayera d’atteindre E de la même façon qui atteint C, par exemple, en envoyant une requête ARP et en attendant la réponse. Quoiqu’il en soit, comme E est physiquement sur un fil différent, il ne verra jamais la requête ARP et le paquet ne pourra pas être délivré. En spécifiant que E est sur un réseau différent, le module IP de A saura que E ne peut être atteint sans avoir été fait suivre par un nœud (élément reliant deux réseaux différents comme un routeur) de son réseau.

# Adresse IP

## Adresse IP :

La version actuelle de ce protocole désormais quasi universel repose en partie sur la notion d'adresse **IP** (Internet Protocol) décernée de façon unique pour chaque élément matériel faisant partie d'un réseau

Ces adresses sont codées sur 32 bits, est sont représentées sous la forme de 4 nombre compris entre 0 et 255 (valeur d'un octet) et séparés par un point, soit (par exemple)

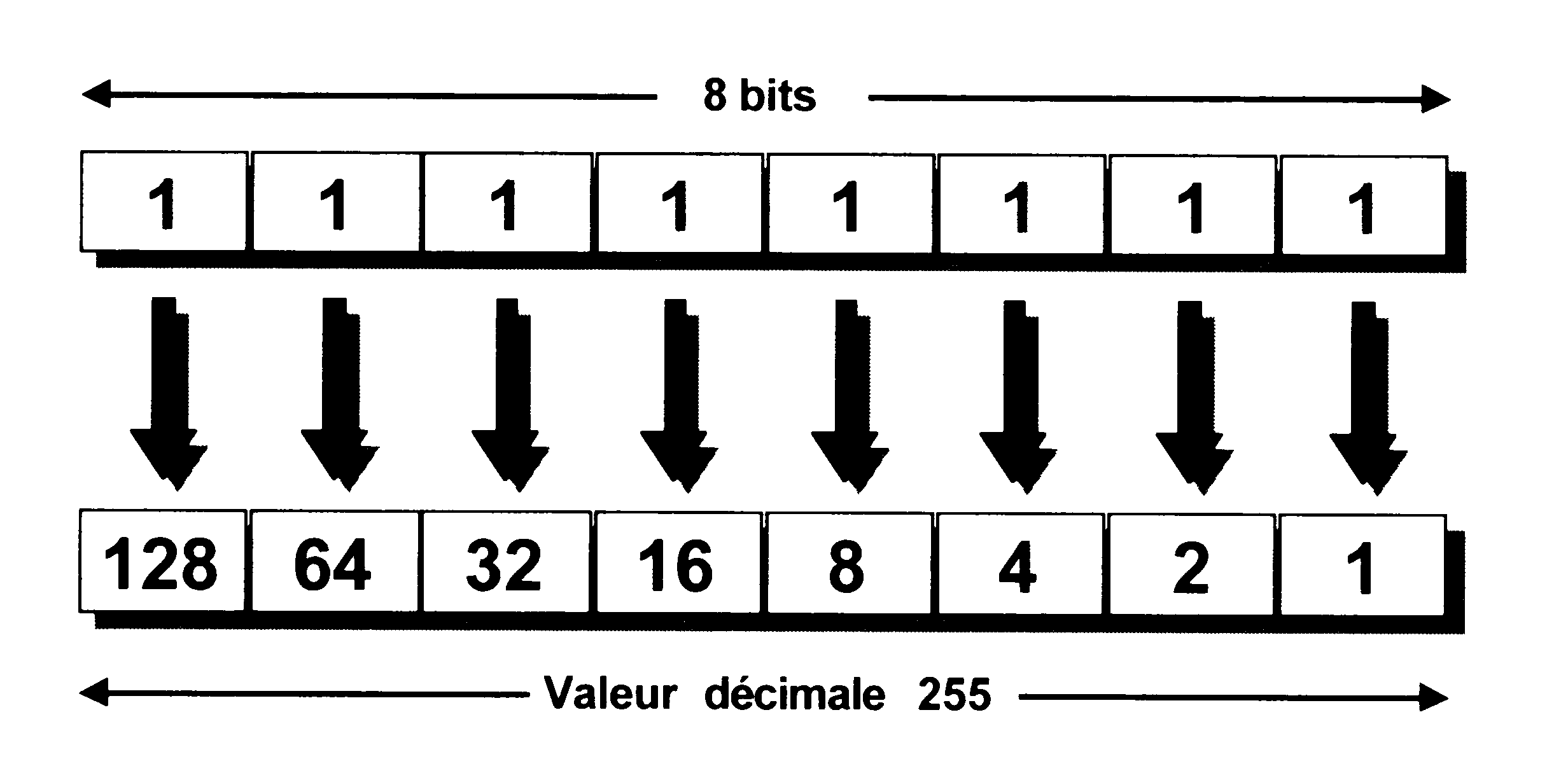
**255.0.127.2**

(ADRESSE DECIMALE)

On parle de bits de poids fort pour les nombres les plus à gauche

On parle de bits de poids faible pour les nombres les plus à droite

Chaque nombre décimal est la représentation d'un nombre binaire de 8 chiffres



On peut alors avoir aussi en notation binaire

(ADRESSE BINAIRE)

**11111111.00000000.01111111.00000010**

On parle de bits de poids fort pour les nombres les plus à gauche

On parle de bits de poids faible pour les nombres les plus à droite

On pourrait ainsi dire que les adresses IP varient de la plus petite 0.0.0.0 à la plus grande 255.255.255.255

**N.B**: En fait toutes les combinaisons ne sont pas disponibles, et elles reflètent une certaine logique

## ID réseau et ID hôte :

Les bits de poids fort définissent l'adresse du réseau, on parle de **ID réseau** et Les bits de poids faible définissent l'adresse d'un équipement dans le réseau on parle de **ID hôte**.

L' **ID réseau** identifie toutes les machines qui se trouvent sur le même réseau physique , encore appelé domaine de collision. Il s'agit d'un identifiant pour un réseau local, toutes les machines se trouvant "du même côté d'un routeur…(sur la même « patte »…)

L' **ID hôte** identifie tout poste ou périphérique du réseau, il est unique à l'intérieur de tout **ID réseau**

## Classes d'Adresse :

La limite entre poids fort et poids faible n'est pas toujours la même, c'est la notion de "**classe d'adresse**"

1. plus les poids fort sont petits, et plus le nombre de machines dans un même réseau sera important, même si on aura peut de réseau
2. plus les poids fort sont nombreux, on aura alors peut de machines connectable pour chacun de ces réseau, même s'il sont plus nombreux

Réseau de **Classe A** : (commence par **1** à **127**)

**0XXXXXXX.XXXXXXXX.XXXXXXXX.XXXXXXXX**

7 bits pour le réseau soit 27=127réseaux

1° bit à 0

24 bits pour le matériel soit 224 (16 millions de machines)

Réseau de **Classe B** : (commence par **128** à **191**)

**10XXXXXX.XXXXXXXX.XXXXXXXX.XXXXXXXX**

14 bits pour le réseau soit 214=16348 réseaux

16 bits pour le matériel soit 216 (65536 machines)

1° bit à 1

2° bit à 0

Réseau de **Classe C** : (commence par **192** à **223**)

**110XXXXX.XXXXXXXX.XXXXXXXX.XXXXXXXX**

8 bits pour le matériel soit 28 (256 machines)

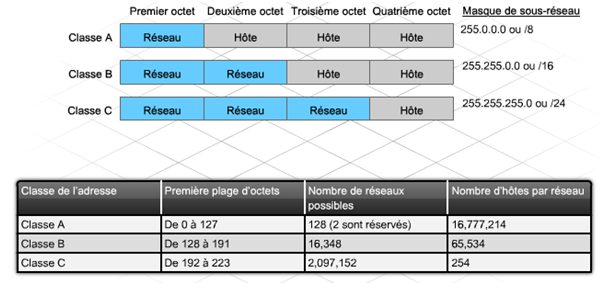
21 bits pour le réseau soit 221=2097152 réseaux

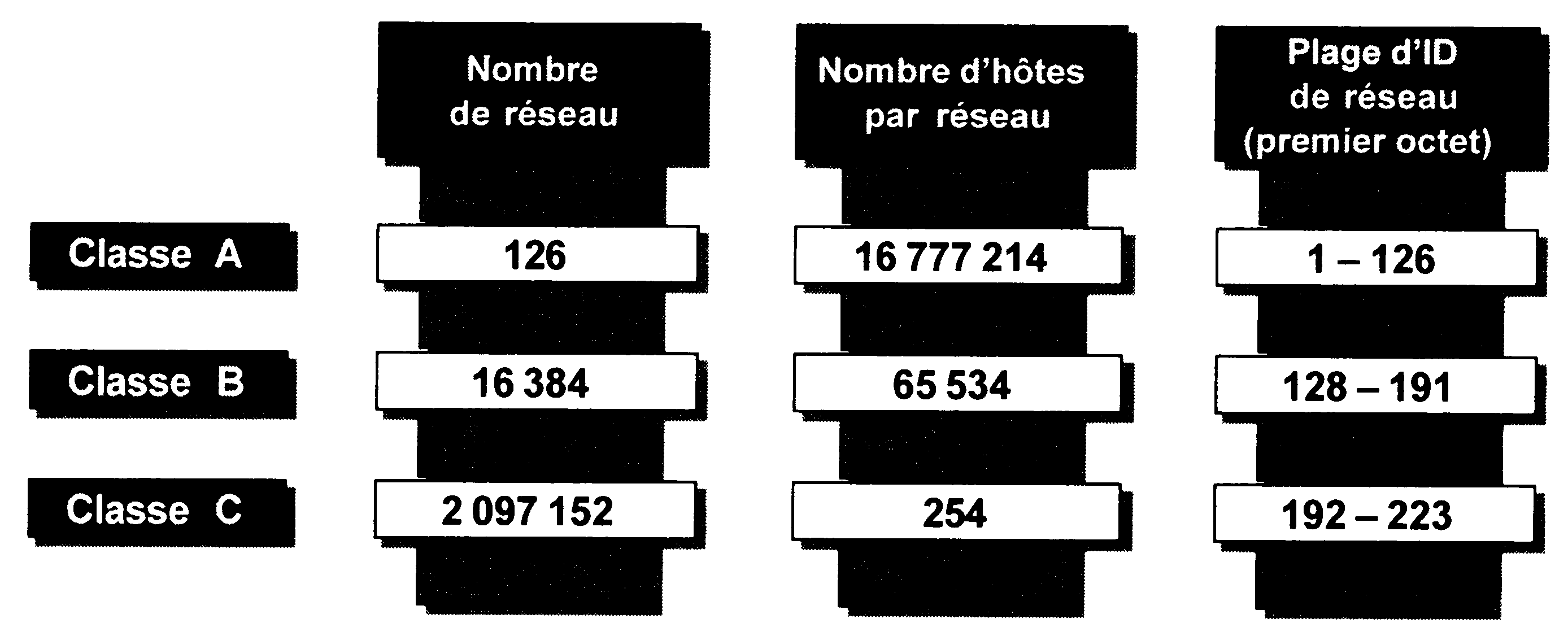
1° bit à 1

2° bit à 1

3° bit à 0

Soit en résumant





Avec quelques règles supplémentaires :

* l'**ID réseau 127**, est réservée pour les tests
* Un **ID réseau** composé exclusivement de 1 ou de 0 n'est jamais attribué
* Un **ID hôte** composé exclusivement de 1 ou de 0 n'est jamais attribué
* La valeur **255.255.255.255** correspond à une diffusion générale (**Broadcast**)

## Adresses IP Privées :

Il est normal d'assigner des adresses globalement uniques à toutes les machines qui utilisent TCP/IP.

Les machines qui utilisent TCP/IP peuvent être divisées en 3 catégories:

* **Catégorie 1 :** les machines qui n'ont pas besoin d'accéder à des machines d'autres entreprises ou à l'Internet dans son ensemble. Les machines de cette catégorie peuvent utiliser des adresses IP qui sont uniques dans l'entreprise, mais qui peuvent être ambigues entre différentes entreprises.
* **Catégorie 2 :** les machines qui ont besoin d'accéder à un nombre limité de services extérieurs (ex: E-Mail, WWW, FTP) qui peuvent êtres servis par des passerelles applicatives. Pour beaucoup de machines dans cette catégorie, un accès non restreint (fourni par la connectivité IP) n'est pas forcément nécessaire et même quelque fois non désiré pour des raisons de sécurité. Pour les mêmes raisons que pour les machines de la première catégorie, de telles machines peuvent utiliser des adresses IP uniques dans l'entreprise, mais qui peuvent être ambigues entre différentes entreprises.
* **Catégorie 3 :** les machines qui ont besoin d'un accès réseau à l'extérieur de l'entreprise (fourni par la connectivité IP). Les machines de cette dernière catégorie ont besoin d'une adresse unique sur tout l'Internet.

On parle pour les machines des catégories 1 et 2 comme de machines "privées", et pour les machines de la 3eme catégorie comme des machines "publiques".

L'Autorité d'Affectation de Numéros sur Internet a réservé les 3 blocs suivant dans l'espace d'adressage pour des réseaux internes RFC 1918:

le premier bloc n'est rien d'autre qu'une classe A n° **10.**

**10.0.0.0 - 10.255.255.255 (10/8 prefix)**

le second, un ensemble de 16 classes B contiguës entre n° **172.16. et 172.31.**

**172.16.0.0 - 172.31.255.255 (172.16/12 prefix)**

N.B: pour **172.16.0.** le premier hôte dispo sera .0.1 (éviter N° à 0 totalement)

et le troisième, un ensemble de 256 classes C de n° **192.168.0. à 192.168.255.**

**192.168.0.0 - 192.168.255.255 (192.168/16 prefix)**

N.B: pour **192.168.0.** le premier hôte dispo sera .1 (éviter N° à 0 totalement)

Les **machines privées** peuvent communiquer avec toutes les autres machines de l'entreprise, à la fois publiques et privées. Néanmoins, elles ne peuvent avoir de connectivité IP avec une machine à l'extérieur de l'entreprise. Même si elles n'ont pas de connectivité IP vers l'extérieur, les machines privées peuvent toutefois avoir accès à des services extérieurs grâce à des passerelles (ex passerelles applicatives).

Pour connecter un réseau utilisant des adresse privées RFC 1918 sur internet, il est nécessaire de prévoir un système de traduction d'adresse (Network Address Translator) ou un système de proxy

Les **machines publiques** peuvent communiquer avec d'autres machines privées ou publiques à l'intérieur de l'entreprise et possèdent une connectivité IP avec les machines publiques extérieures à l'entreprise. Les machines publiques n'ont pas de connectivité avec des machines privées d'autres entreprises.

# masque de sous-réseau

## Subdivision de réseau :

Très fréquemment on constitue un réseau à partir de segments ou brins interconnectées entre eux via des routeurs…

Les avantages à avoir un réseau bien segmenté sont nombreux :

* Différentes techniques de réseau peuvent être mélangées (Ethernet et Token-Ring par exemple…)
* Les collisions sont limitées car les diffusions générales sont limitées au segment local
* Extension à un nombre pratiquement infini d'hôtes

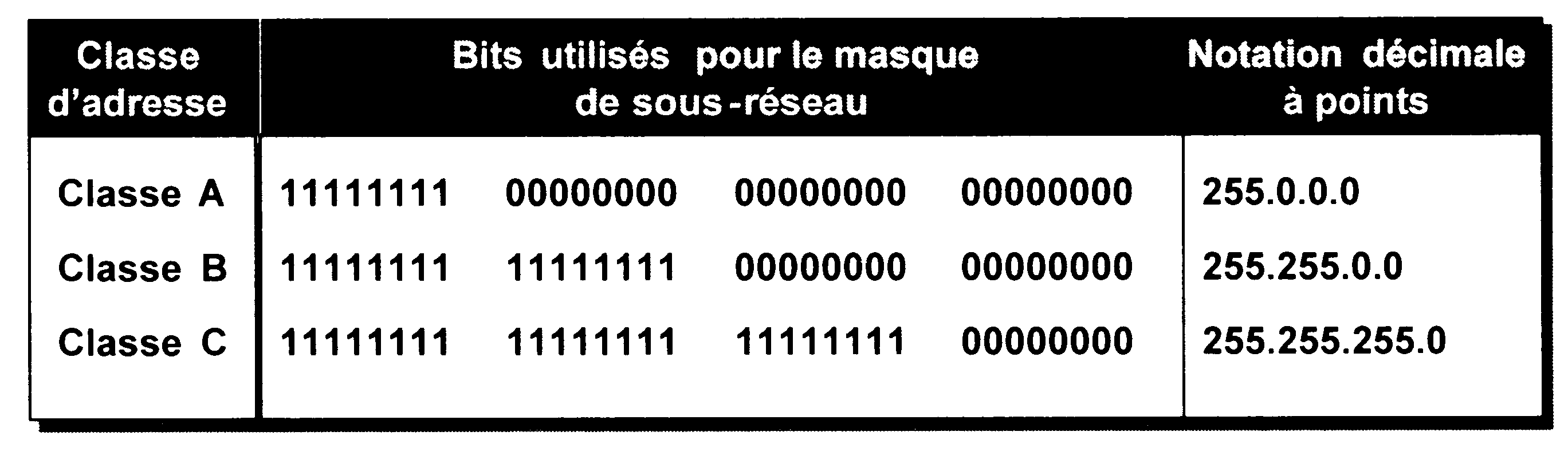
## Masque de sous-réseau :

Le **masque de sous-réseau** permet de définir le découpage entre les bits de l'adresse qui servent à définir l'adresse de réseau, et ceux servant à définir l'adresse de la machine

En effet via un système de **ET bit à bit**, le **masque de sous-réseau** permet de distinguer l'**ID réseau** à partir de l'**Id hôte**, et par conséquent permet à **TCP/IP** de savoir si une **adresse IP** donnée se trouve sur le **réseau local** ou sur un **réseau distant**

## Masque par défaut :

Ainsi dans des masques standards, tous les bits correspondants à l'**ID réseau** sont à 1, tous les bits correspondants à l'**ID hôte** sont à 0



## Masque personnalisé :

L'objectif est ici d'obtenir des adresses d'**ID réseau** et d'**Id hôte** groupées de manière un peu différente par rapport aux classes standardisées A-B-C qui servent de cadre

Pour définir des sous-réseaux personnalisés, il est nécessaire de définir :

* Combien de réseau veut on gérer à l’intérieur de la plage d'adresse attribuée
* Combien d'hôtes maximum veut on gérer à l'intérieur d'un sous-réseau

N.B: en prévoyant une évolution future raisonnable !

Puis travailler de la manière suivante :

* Définir le masque de sous-réseau qui donne le nombre de sous-réseau et d'hôte par sous-réseau voulu
* Déterminer les **ID réseaux** possibles à utiliser   
  N.B: ( cf tables page 24 pour savoir combien il y en a)
* Déterminer les **ID hôtes** possibles à utiliser   
  N.B: ( cf tables page 24 pour savoir combien il y en a)

### Définir un masque de sous-réseau

On l'a dit, l'**ID réseau** se calcule en regardant le nombre de 1 du masque de sous-réseau.

Pour augmenter le nombre d'**ID réseau**, il faut ajouter des bits au masque de sous-réseau (Bien sûr si on augmente le nombre d'**ID réseau**, on diminue le nombre d'**ID hôte**…)

**De combien de bit faut-il augmenter le masque de sous-réseau ?**

Comme on travaille avec les puissances de 2, on augmente les combinaisons de 2 ^ nb bits ajoutés

soit 1 bit 2 sous-réseaux

2 bit 4 sous réseaux

3 bits 8 sous réseaux

4 bits 16 sous réseaux

5 bits 32 sous réseaux

x bits 2^x. sous-réseaux

mais rappelez vous, les adresse ne contenant que des 0 ou que des 1 ne sont pas autorisées, par conséquent il faut enlever les 2 adresses extrêmes possibles…ce qui nous donne

soit **1 bit** **impossible** (2-2=0)

**2 bit** **2 sous réseaux** (4-2)

**3 bits** **6 sous réseaux** (8-2)

**4 bits** **14 sous réseaux** (16-2)

**5 bits** **30 sous réseaux** (32-2)

x bits (2^x)-2 sous-réseaux

**Comment calculer le nouveau masque de sous-réseau de mes réseaux?**

1. Une fois trouvé le nombre de bits me permettant d'obtenir le nombre de sous-réseaux voulu, je dois créer un octet avec :

**a gauche** les bits trouvés auparavant

a droite des *0* permettant de "compléter l'octet"

6 ss reseaux 2 ss réseaux 30 ss réseaux

3 bits + 5 zéro 2 bits + 6 zéro 5 bits + 3 zéro

**111***00000* **11***000000* **11111***000*

1. puis le convertir en décimal

**111***00000* **11***000000* **11111***000*

=128+64+32 =128+64 =128+64+32+16+8

224 192 248

1. et remplacer dans la masque par défaut de ma classe d'adresse, le premier 0 par ce nombre…

6 ss reseaux 2 ss réseaux 30 ss réseaux

224 192 248

si l'adresse est de classe **A** cela donne par rapport au masque **255.0.0.0**

**255**.224.0.0 **255**.192.0.0 **255**.248.0.0

si l'adresse est de classe **B** cela donne par rapport au masque **255.255.0.0**

**255.255**.224.0 **255.255**.192.0 **255.255**.248.0

si l'adresse est de classe **C** cela donne rapport au masque **255.255.255.0**

**255.255.255**.224 **255.255.255**.192 **255.255.255**.248

**Comment calculer les ID réseau de mes réseaux?**

1. Recenser toutes les combinaisons possibles (en excluant donc celles n'ayant que des 1 ou des 0) de bits ajoutées au masque de sous-réseau précédemment et les convertir en décimal:

6 ss réseaux 2 ss réseaux 30 ss réseaux

**(111)***00000* **(11)***000000* trop long !

**110***00000* **10***000000*

**101***00000* **01***000000*

**011***00000* **(00)***000000*

**100***00000*

**010***00000*

**001***00000*

**(000)***00000*

1. Les convertir en décimal:

6 ss réseaux 2 ss réseaux

192 64

160 32

128

96

64

32

1. Ajouter ces valeurs a l'**ID réseau** d'origine:

**Comment calculer les ID hôtes disponibles dans mes réseaux?**

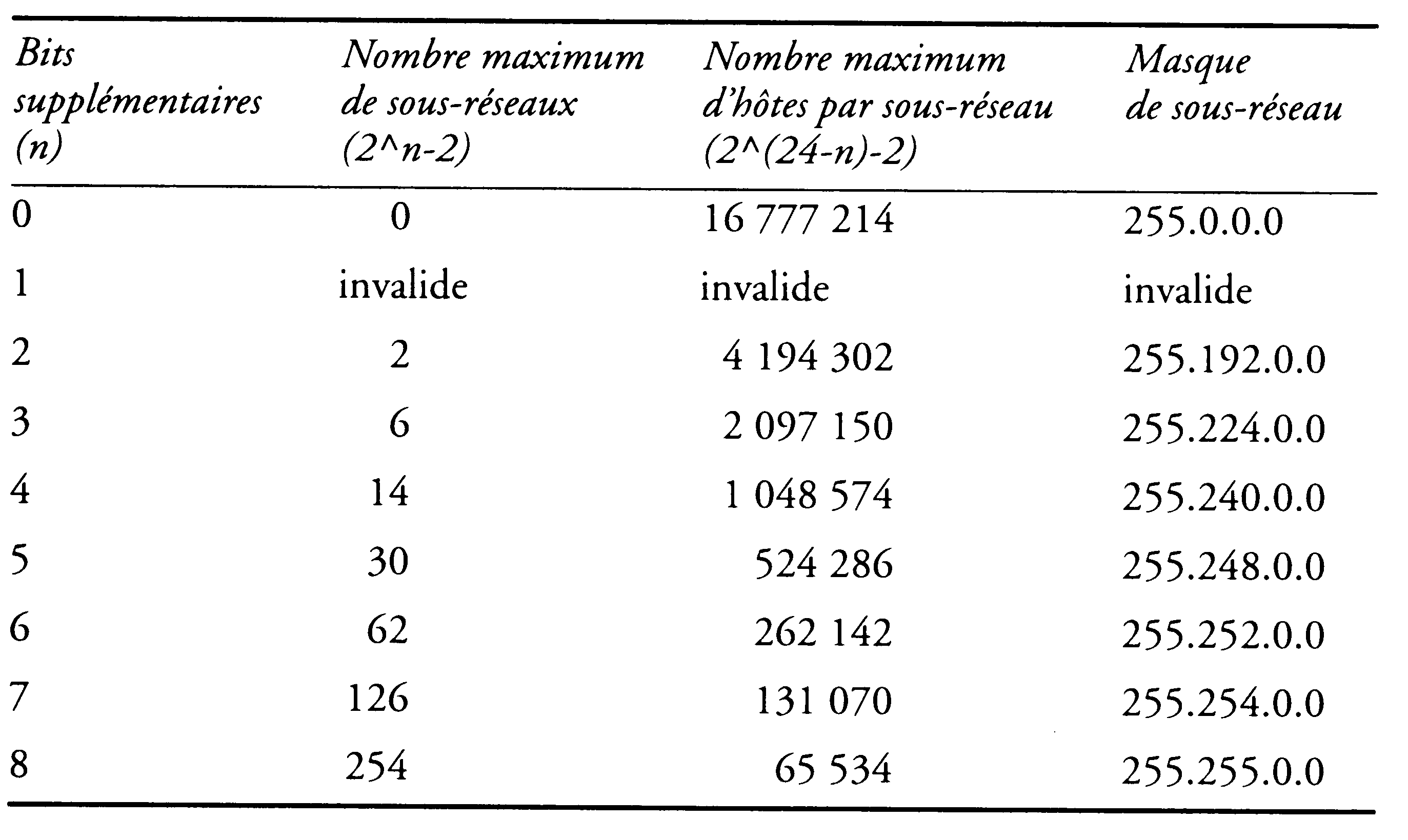
Les **ID hôte** commencent par la valeur .001 dans le dernier octet et augmentent 1 par 1 jusqu'à atteindre la valeur ID de sous-réseau du réseau suivant, -1

Bien sûr le dernier octet lui aussi ne peut pas être égal à 0 ou 255.

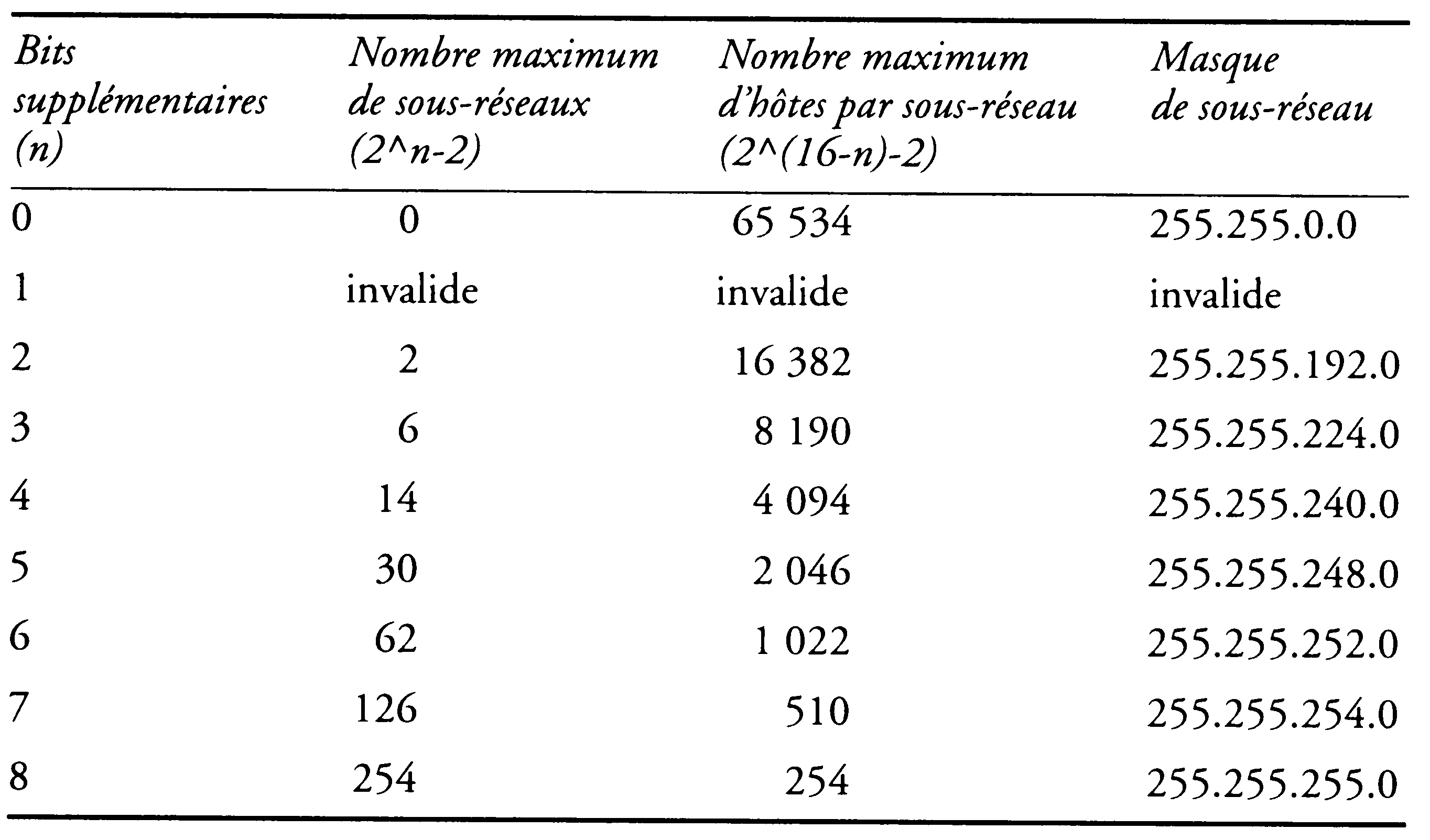
## Tables de définition des sous-réseaux :

Voilà le nombre de sous-réseau utilisables, avec le nombre d'hôte possible pour un masque de sous-réseau donné, et ce pour les

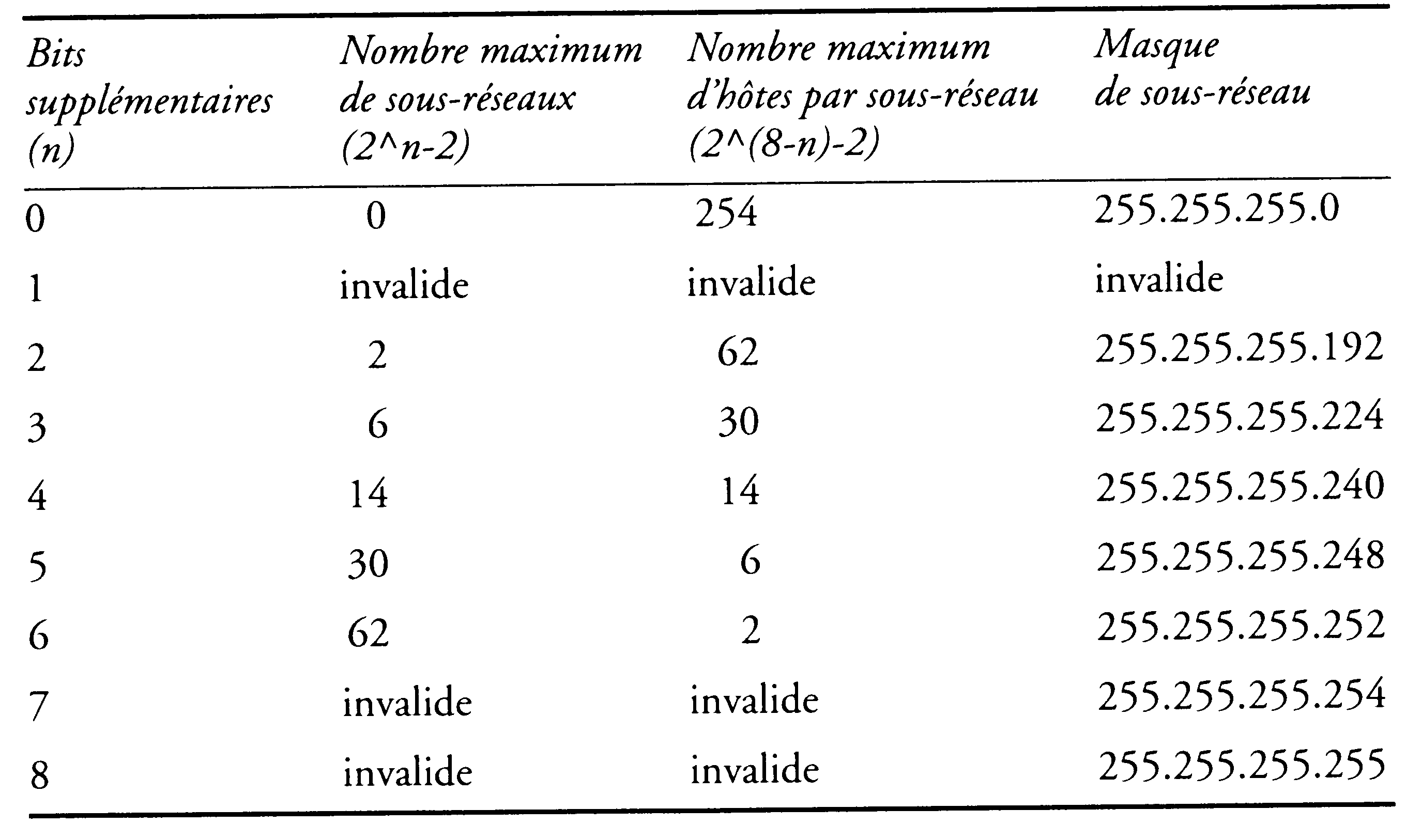
**Adresses de classe A:**



Adresses de classe B:



Adresses de classe C:



### Exemple 6 sous réseaux de 30 postes :

Si on veut **6 sous réseaux** comportant chacun 30 machines maximum, on pourra prendre alors comme masque de sous réseau **255.255.255.224**

* **Id réseau**pour trouver les Id réseau je dois trouver toutes les combinaisons de **3 bits** de 111 à 000 en laissant tomber les valeurs n'ayant que des 0 ou que des 1 (non autorisée).J'obtient 110-101-011-100-010-001 soit en décimal 192-160-128-96-64-32.

que je rajoute à mon Id réseau d'origine 192.168.1.xx soit donc les Id réseau suivantes :

192.168.1.**192** 192.168.1.**160** 192.168.1.**128** 192.168.1.**96**

192.168.1.**64** 192.168.1.**32**

* Id hôte valide   
  un petit calcul nous donne :

**sous-réseau 1° adresse IP dernière adresse IP**

192.168.1.**32** 192.168.1.33 192.168.1.63

192.168.1.**64** 192.168.1.65 192.168.1.95

192.168.1.**96** 192.168.1.97 192.168.1.127

192.168.1.**128** 192.168.1.129 192.168.1.159

192.168.1.**160** 192.168.1.161 192.168.1.191

192.168.1.**192** 192.168.1.193 192.168.1.223

# masque de sur-réseau

## Objectif du sur-réseau :

La question ici n'est pas de délimiter des sous-réseaux (donc des sous-ensemble de moins de 255 machines pour une classe C par exemple), **mais plutôt de faire en sorte que l'on puisse adresser "ensemble" plus de 255 machines, mais en restant avec des adresses de classe C ! (par exemple)**

Ainsi imaginons un réseau constitué au départ d'une centaine de machines dont les adresses IP privées ont étés définies en classe C, par exemple sur les adresses de base suivantes: 192.168.25.1 à 192.168.25.100. Ce réseau grandit, et voit le nombre des machines dépasser les 255 postes, que faire ?

classiquement on peut agir de différentes manières :

* Fractionner le réseau en plusieurs zones distinctes, et les relier par un (des) routeurs...
* Passer à des adresse de type Classe B, par exemple 172.16.0.1 à 172.16.1.xxx avec un masque par défaut de 255.255.0.0
* Augmenter la taille du masque par défaut, de 255.255.255 à ....  
  c'est du sur-réseau !

## Principe :

l'agrégation de plage d'adresse, ou "**super-netting**" s'effectue en modifiant le masque de sous-réseau. La modification, dépends du nombre (puissance de 2)de classe que l'on souhaite "agréger" :

|  |  |  |
| --- | --- | --- |
| **Nombre Classes à agréger** | **Masque sous-réseau** | **nombre de Hosts maximum disponibles** |
| 1 | 255.255.255.0 | 256 |
| 2 | 255.255.254.0 | 512 |
| 4 | 255.255.252.0 | 1024 |
| 8 | 255.255.248.0 | 2048 |
| 16 | 255.255.240.0 | 4096 |
| 32 | 255.255.224.0 | 8192 |
| 64 | 255.255.192.0 | 16384 |
| 128 | 255.255.128.0 | 32768 |
| 256 | 255.255.0.0 | 65536 |

Dans notre cas pour adresser un maximum de 1024 machines, il faut agréger 4 classes par exemple, et comme masque prendre la valeur 255.255.252.0,

Ce qui permet d'avoir en fait 256/4 plages adressables de 1024 machines chacune, suivant le tableau ci-dessous :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **N° plage** | **Adresse Début** | **Adresse Fin** | **Masque** | **nb Hosts maxi** |
| 1 | 192.168.0.0 | 192.168.3.255 | 255.255.252.0 | 1024 |
| 2 | 192.168.4.0 | 192.168.7.255 | 255.255.252.0 | 1024 |
| 3 | 192.168.8.0 | 192.168.11.255 | 255.255.252.0 | 1024 |
| 4 | 192.168.12.0 | 192.168.15.255 | 255.255.252.0 | 1024 |
| 5 | 192.168.16.0 | 192.168.19.255 | 255.255.252.0 | 1024 |
| ...x... | 192.168. (x\*4)-4 .0 | 192.168. (x\*4)-1 .255 | 255.255.252.0 | 1024 |
| 64 | 192.168.252.0 | 192.168.255.255 | 255.255.252.0 | 1024 |

**N.B**: les adresses faisant partie du même N° plage sont vues comme faisant partie d'une même réseau, donc ne nécessitent pas de routage entre elles

**N.B**: les adresses ne faisant pas partie du même N° plage sont vues comme faisant partie de réseaux différents, donc nécessitent un routage entre elles

# le Routage TCP/IP

## Notion de routeur :

De manière générale, une machine peut communiquer uniquement par défaut avec une autre machine de son réseau local, c'est à dire une autre machine faisant partie de son sous-réseau, encore appelé domaine de collision.

Que ce sous-réseau soit obtenu par l'application d'un masque de sous-réseau par défaut isolant des classes A, B ou C complète, ou qu'il soit obtenu par l'application d'un masque de sous-réseau personnalisé modifiant l'étendue par défaut des ID réseau et des ID hôtes, l'idée est la même :

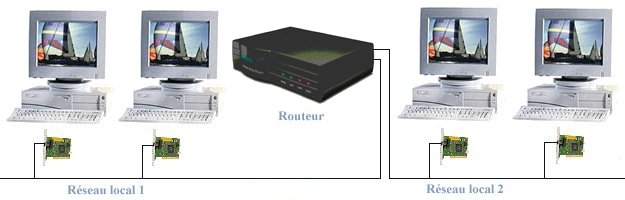
IP compare l'ID de sous-réseau de l'adresse IP que l'on cherche à joindre à l'ID de sous-réseau du réseau local dans lequel il se trouve :

* **Si les deux ID correspondent** : IP peut chercher localement la machine
* **Si les deux ID ne correspondent pas** : IP envoi la trame vers un équipement ou il peut être routé

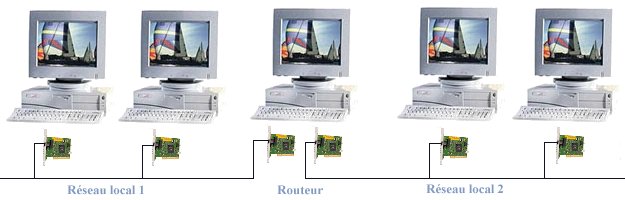
Lorsque des machines sont interconnectées en plusieurs sous-réseaux, elles doivent toutes avoir comme paramétrage l'adresse IP d'une passerelle - **routeur** par défaut

Une adresse IP différente est assignée a chaque carte sur chaque sous-réseau, permettant à ce **routeur** de faire partie de plusieurs réseaux différents. On parle alors aussi **d'hôte multi-résident**.

Un routeur peut être soit un matériel spécifique,



Soit une fonction assurée par une station de travail possédant au moins deux interfaces réseaux, et une application pour le routage.

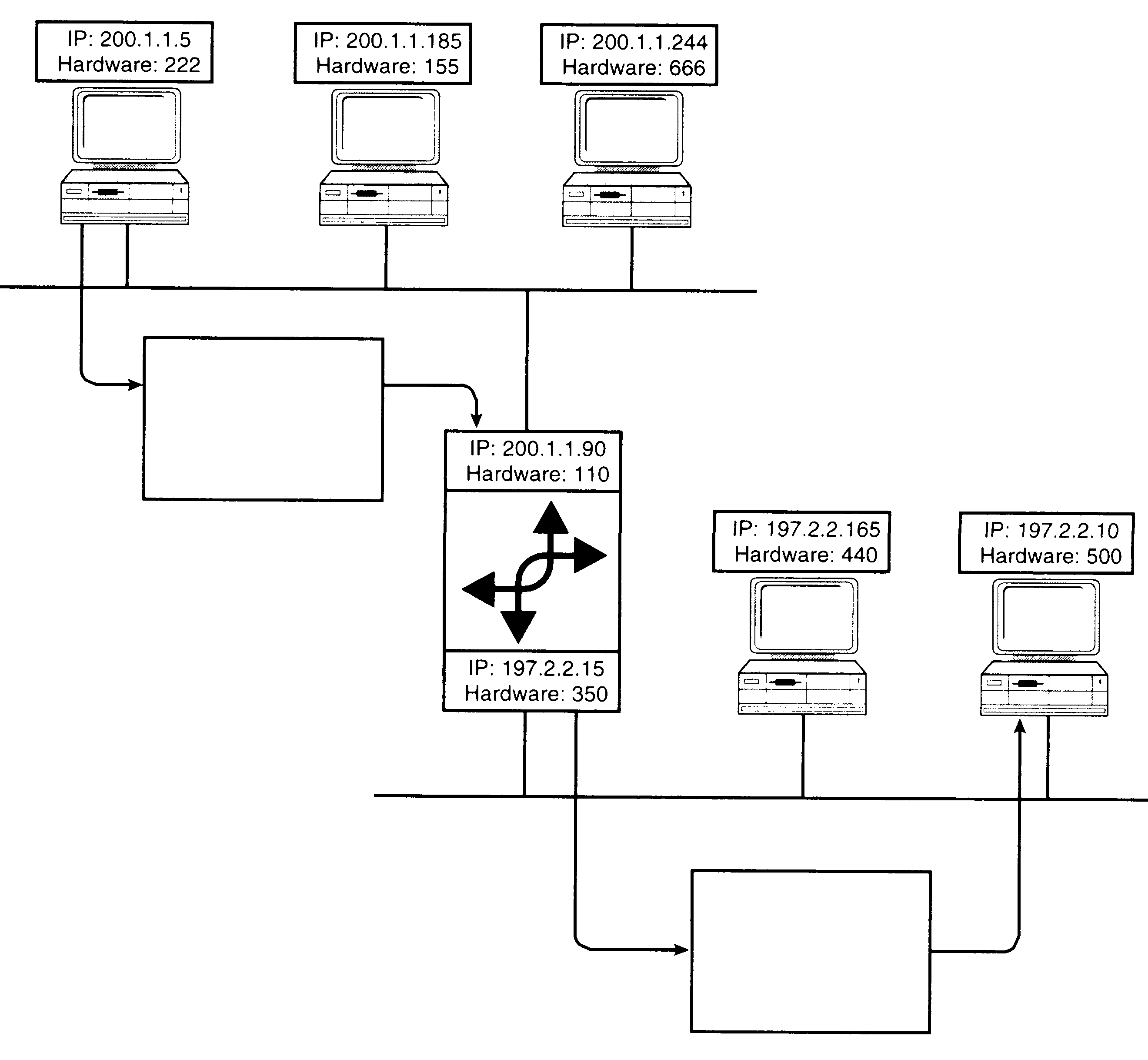


**N.B:** Toute machine windows Serveur peut faire office de routeur à partir du moment ou elle dispose d'autant de cartes réseaux que de sous-réseaux auxquels elle souhaite être rattachée.

## Routage de base :

Dans la situation la plus simple, on relie **deux sous-réseaux** par **un routeur** ayant donc **deux cartes réseaux** et **deux adresses IP** dans chaque sous-réseau auxquels il appartient :

Dans l'exemple ci-dessous, la machine d'adresse **IP 200.1.1.5** essaye de joindre la machine d'adresse **IP 197.2.2.10**…



vers passerelle

vers machine

Le fonctionnement est le suivant :

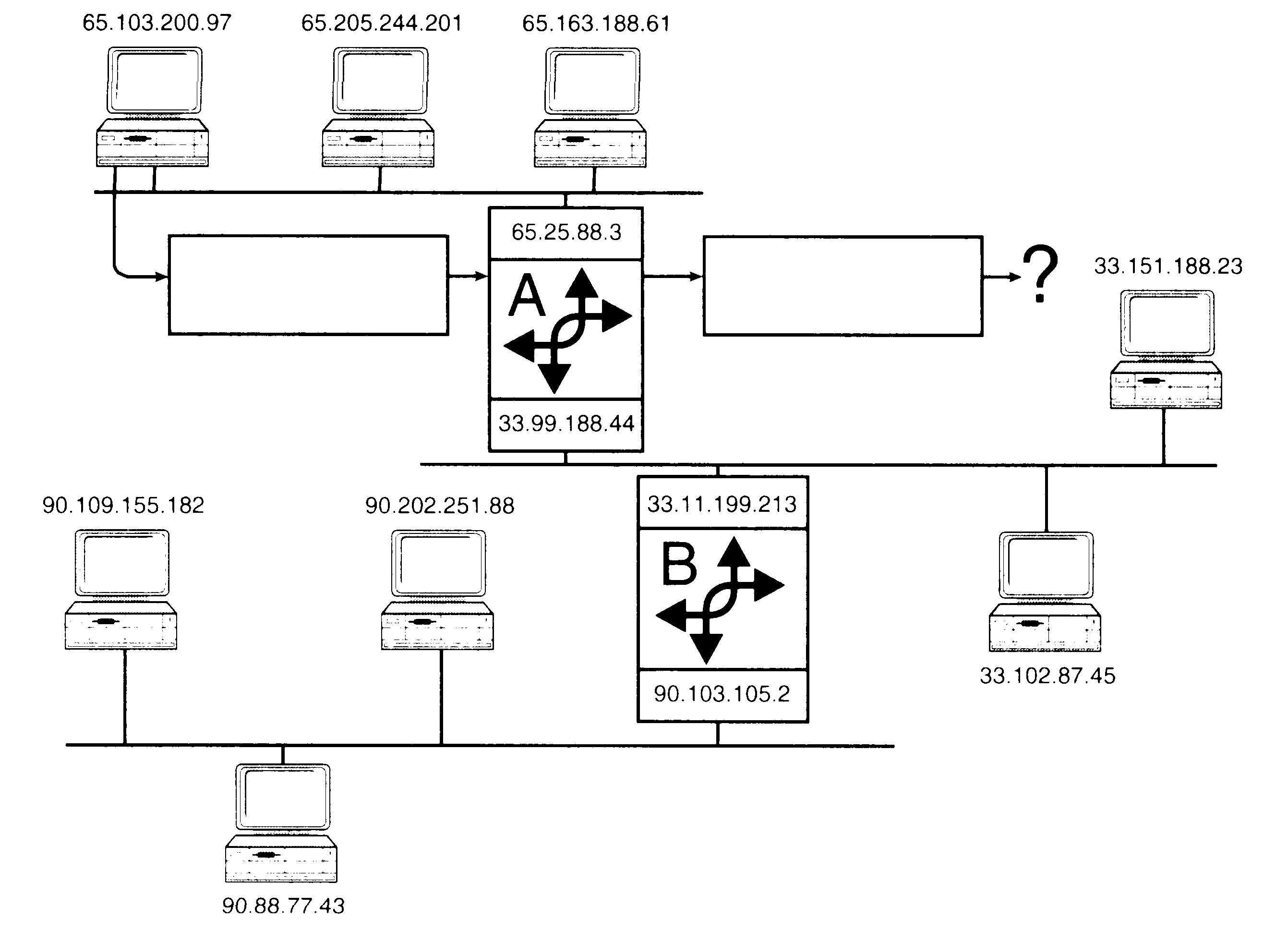
1. IP se rends compte que l'adresse de destination n'est pas une adresse locale (ID réseau cherchée **197.2.2.0** différente de ID réseau locale **200.1.1.0**)
2. IP transmet alors le paquet à la passerelle par défaut
3. IP sur le routeur détermine que l'ID cherchée est **197.2.2.0**, comme le routeur possède une carte paramétrée sur ce réseau il l'utilise pour envoyer ce paquet...
4. IP sur la machine de destination récupère le paquet qui lui est destiné…

**N.B:** Par défaut, les tables de routage sur Windows ne contiennent que des informations sur les sous-réseaux sur lesquels le routeur est directement connecté. Ce qui est sans doute un peu limitatif…

## Routage complexe :

Dans une situation plus complexe, on relie **trois sous-réseaux** **par deux routeurs** ayant chacun **deux cartes réseaux** et **deux adresses IP** dans chaque sous-réseaux auxquels ils appartiennent :

Dans l'exemple ci-dessous, la machine d'adresse **IP 65.103.200.97** essaye de joindre la machine d'adresse **IP 90.109.155.182**…



➋

➊

Le fonctionnement est le suivant :

1. IP se rends compte que l'adresse de destination n'est pas une adresse locale (ID réseau cherchée **90.0.0.0** différente de ID réseau locale **65.0.0.0**), IP transmet alors le paquet à la passerelle par défaut
2. IP sur le routeur détermine que l'ID cherchée est **90.0.0.0**, mais comme le routeur **ne possède pas** une carte paramétrée sur ce réseau il ne sait pas où envoyer ce paquet...

**N.B:** Par défaut, les tables de routage sur Windows NT ne contiennent que des informations sur les sous-réseaux sur lesquels le routeur est directement connecté, ce qui fait que ici le paquet ne saurait être routé vers le réseau 90.0.0.0

## Table de Routage :

Dans une situation plus complexe, il est nécessaire de configurer un routeur avec une table de routage qui contient des informations destinées a router des paquets vers d'autres routeurs lorsque l'on ne sait pas directement qui pourrait les prendre en charge.

Dans notre exemple il faudrait indiquer à notre premier routeur que lorsqu'il reçoit des paquets à destination d'un réseau 90 il doit les router vers le réseau 33.0.0.0

Dans notre exemple toujours, le cas n'étant que peu compliqué, on pourrait s'en sortir en paramétrant comme passerelle par défaut de ce routeur, l'adresse du deuxième routeur…

Cette méthode est limité au cas où l'on a que 2 routeurs …

D'une manière plus générale il va falloir configurer une table de routage…

## Routage statique :

On appelle **routage statique** un routage qui est mis à jour manuellement sur chaque routeur par l'administrateur

La commande permettant de créer et maintenir une table de routage est la commande

route print

## Routage dynamique :

On appelle **routage dynamique** un routage qui est mis à jour automatiquement sur chaque routeur par échange d'information entre les routeurs…

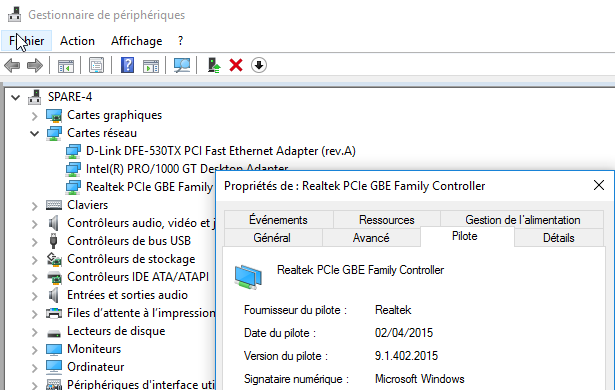
**N.B**: Windows ne dispose pas de cette capacité à travers le protocole RIP

# Réseau Windows 10

## Gestion Carte Réseau:

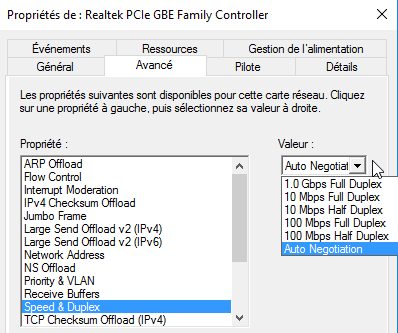
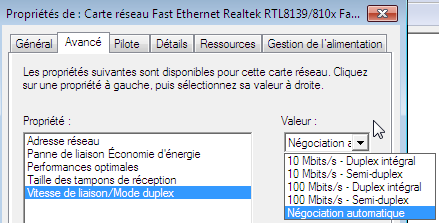
Depuis Windows 10, une myriade d'assistant se déclenchent à tous moments, les interfaces sont assez "fluctuantes" (selon les versions 1511, 1607, 1703, 1709) et "fournies"…

Si aucune carte réseau n'est détectée, il faut installer un driver certifié …



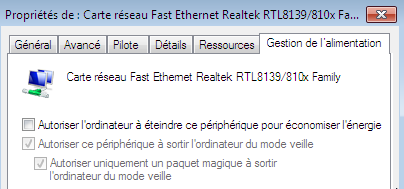
En cas de problème "physiques", on peut vérifier que le driver gère correctement nos flux Ethernet selon notre connectique (et passer en vitesse de remplis si besoin)

Carte **Gb/s** Carte **100Mb/s**



**N.B**: ces réglages sont parfois difficiles à trouver, ils dépendent bien sur des drivers …

On peut aussi éviter pour des raisons ACPI d’éteindre la carte réseau…



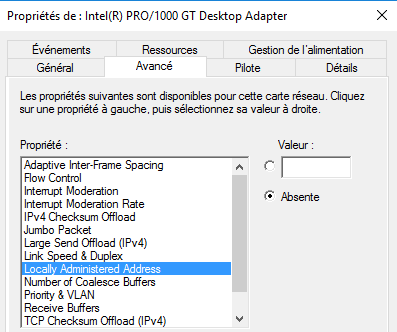
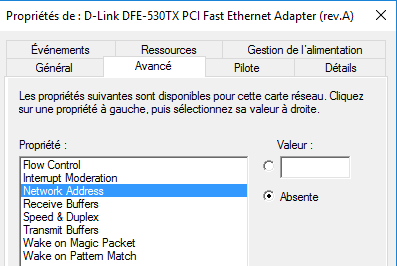
Toutes les cartes n’offrent pas tous les réglages, ni sous les mêmes libellés :

Ainsi pour **Adresse Mac**, **Vitesse + Mode**, **MTU**, on peut trouver par exemple

Locally Administered Address Network Address

Link Speed & Duplex Speed & Duplex

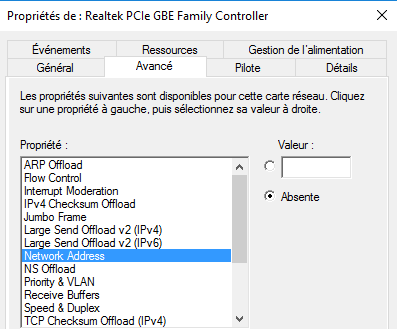
Jumbo packet - non disponible



Network Address

Speed & Duplex

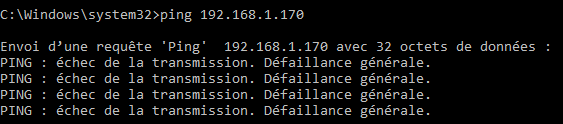
Jumbo Frame



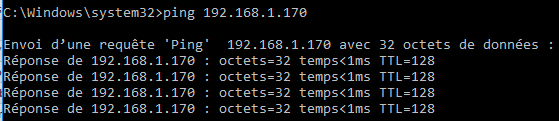
## Désactivation Media Sense:

Windows dispose de la fonction de « Détection de support ». **Media Sense**

Un « état de la liaison » est défini comme étant le support physique connecté ou inséré sur le réseau. Chaque fois que Windows détecte un état « inactif »  sur le support, il supprime les protocoles liés de cette carte jusqu'à ce que l'état détecté soit de nouveau « actif ». Une telle carte, génère en réponse à un ping local (par exemple une application locale qui détecterait la présence d’un réseau) une défaillance générale



Pour que votre carte réseau ne désactive plus IP lors de cette situation, et **réponde sur un ping de l’adressse IP en local**



il faut utiliser en invite de commande la commande **netsh** . On peut voir l’état de la situation **Détection de médias DHCP**, dans la commande

**Netsh interface ipv4 show global**



On désactive la fonctionnalité en IPV4 et IPV6 avec la commande

**Netsh interface ipv4 set global dhcpmediasense = disabled**





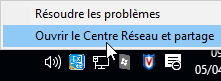
Puis reboot du poste !

## Accès au Centre Réseau et partage :

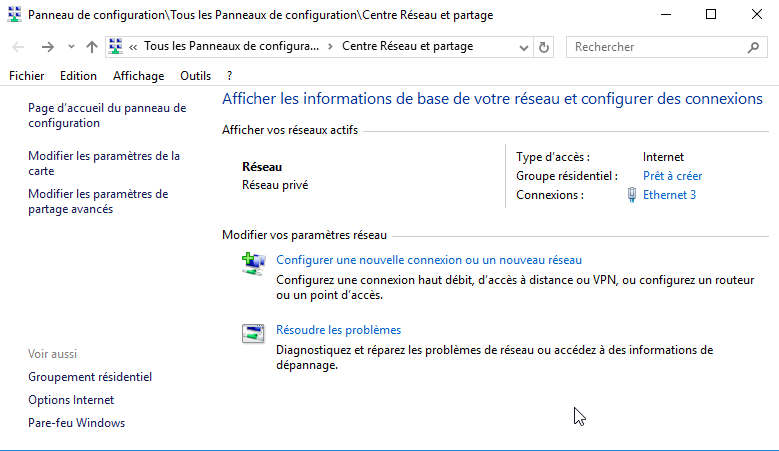
Si une carte réseau (minimum) est installée correctement, une icône "réseau" devrait apparaître en bas à droite… 

Quel que soit son aspect…    

Un clic dessus puis "**Ouvrir le Centre Réseau et partage**"



Devrait amener



On peut aller aussi **Pare-feu Windows**

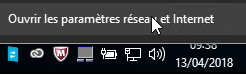
**Modifier les paramètres de partage avancés**

Permet de gérer les paramètres des "profils réseaux". On est depuis windows 10 toujours dans un profil réseau…

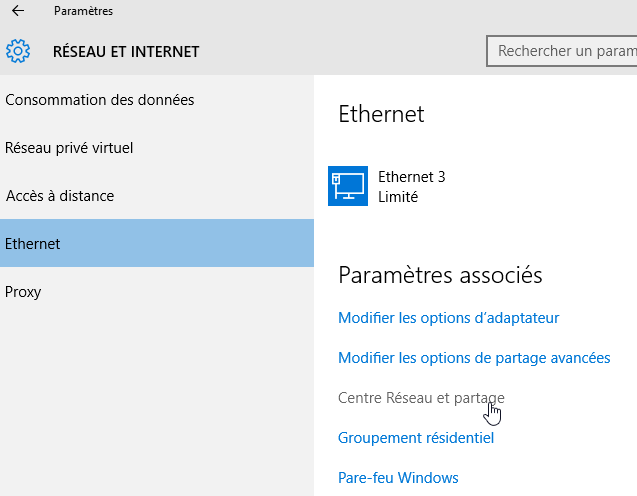
**N.B**: Il existe trois type de "profils" réseau au sens Windows, mais l'utilisateur ne peut choisir que entre **Privé / Public** , car si **Domaine** existe, il ne peut être remis en cause:

**Modifier les paramètres de la carte**

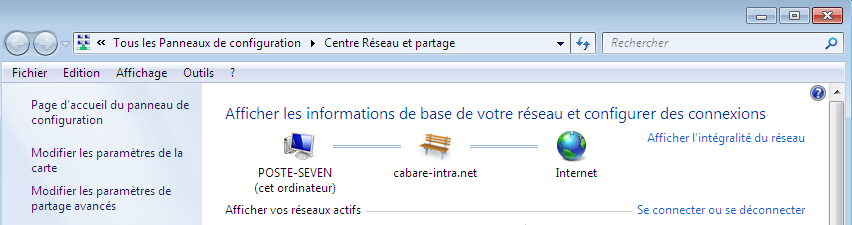
Permet pour chaque carte, de configurer les protocoles, dont TCP-IP…

**N.B** : sous la version **Windows 10 - 1709** la boite de dialogue change, et on n‘accède au **Centre de réseau et partage** qu’après être passé d’abord via **Ouvrir les paramètres réseaux et internet** (équivalent du menu paramètre **Windows 10 / réseau et internet**)

Et on retrouve le **Centre de Réseau et partage** ensuite



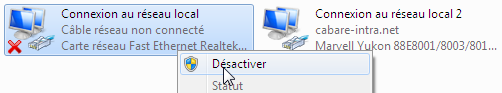
## Désactivation Carte Excédentaire :



**Modifier les paramètres de la carte** donne accès en faità toutes les cartes physiques… Si plusieurs cartes réseaux sont présentes, il est plus judicieux de désactiver celle que l'on n'utilise pas.

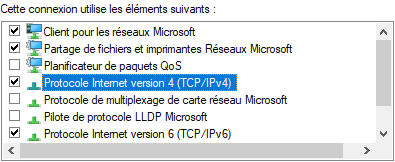


Clic droit **Désactiver**



## Protocoles LLDP - multiplexage - Topologie réseau Windows:

Pour chaque Icône, les propriétés de la carte permettent de définir les services et les protocoles voulus, Lorsque l'on affiche les protocoles disponibles, il y en a beaucoup. Que 4 sont indispensables ! (voire 3)



Protocoles présents (à ne pas activer systématiquement)



**LLDP Link Layer Discovery Protocol** (LLDP) est un protocole 802.1ab. C'est un protocole destiné à remplacer un bon nombre de protocoles propriétaires (Cisco CDP, Extreme EDP, etc.) utilisés dans la découverte des topologies réseau de proche en proche



**Protocole de multiplexage de carte réseau Microsoft** utilisé pour deux scénarios d'utilisation typiques, chacun nécessitant au moins deux adaptateurs réseau fonctionnant (et connectés) sur un même PC. Le premier scénario s'appelle l'association d'adaptateurs, ce qui signifie l'utilisation simultanée de deux adaptateurs (trunk). Le second scénario est appelé basculement de l'adaptateur / haute disponibilité, où un adaptateur de secours prend en charge la connexion réseau en cas d'échec du serveur principal.

Si on veut du voisinage réseau. (cf chap spécifique)

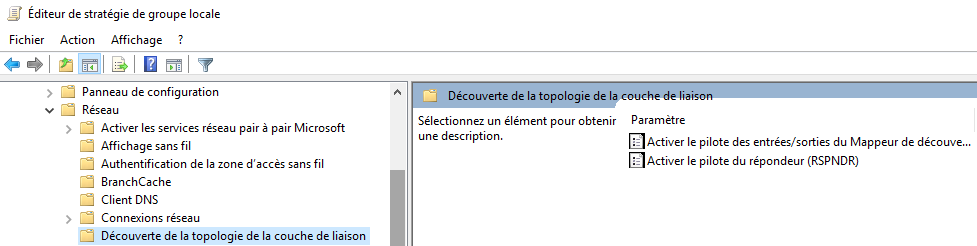


**Pilote E/S de mappage de découverte de topologie de la couche de liaison**

**Répondeur de découverte de la topologie de la couche liaison**

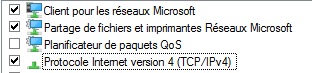
Ces pilote permet de remonter sur une machine tous les partages et les accès sur un réseau local, comme on pouvait l'avoir sous Vista et les premiers Seven. (afficher l'intégralité du réseau). Associé à un répondeur (forcément)

Peut se gérer via **gpedit.msc / modèle d'administration / réseau/ Découverte de la topologie réseau**



## Protocoles Ip-v4 Ip-v6 QoS Client et partage Réseaux

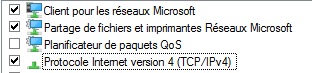
Les 3 protocoles absolument indispensables pour une connexion **ipv4** sont



**Client pour les réseaux Microsoft**

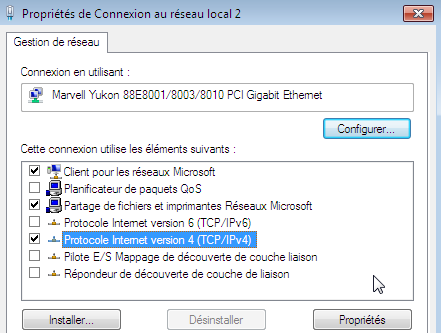
**Partage de fichiers et imprimantes Réseaux Microsoft**

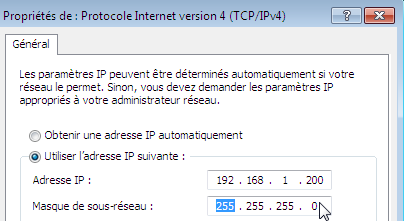
Si on ne les actives pas, on ne pourra rien "faire" avec notre machine.



**Protocole Internet version 4 (TCP/IPv4)**

évidemment





L'adressage de base Minimum réside en une Adresse IP et un Masque.

**N.B**: Pour l'identification du profil réseau, une passerelle facilite les choses



**Planificateur de paquets Qos**

A ne pas activer, sauf si on utilise des applications le nécessitant



**Protocole Internet version 6 (TCP/IPv6)**

Microsoft ne recommande pas la désactivation du protocole **IPv6**. Depuis Windows Vista et Windows Server 2008, IPv6 fait partie intégrante du système d'exploitation. Certains composants utilisent nativement IPv6 : **Remote Assistance** – **DirectAccess** - **Client Side Caching** (offline files) et **BranchCache**

Il ne faut plus prioriser les flux comme on pouvait tenter de le faire sous Windows Seven ou 2008R2 ! Il faut laisser le protocole IPVv6 en client DHCP v6 et ce depuis la version Windows 10 1607.

## Ré-initialiser TCP/IP Sous Windows 10 :

Dans des cas extrêmes, pour réinitialiser la « Pile Ip » on ne peut plus désinstaller le protocole TCP-IP dans l’interface, mais on peut passer une commande en invite de commande.

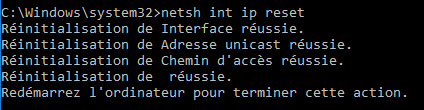
Il faudra impérativement

* redémarrer le poste et
* reprendre toutes les configurations réseaux existantes

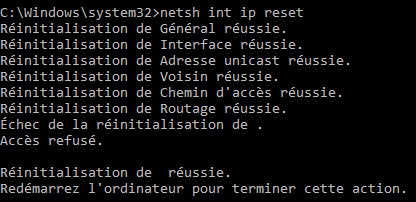
Donc en Invite de commande

**netsh int ip reset**

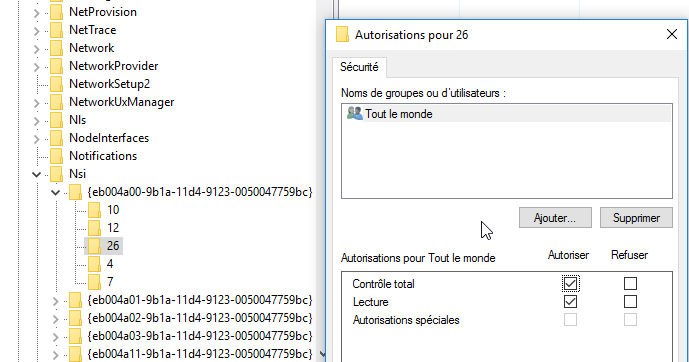
devrait donner



**N.B :** Il se peut que l’on ait un petit souci sur la composante DHCP, en Workgroup

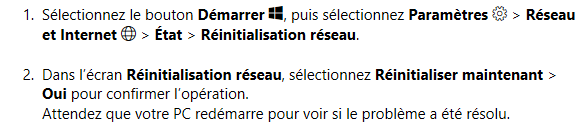


Cela peut se résoudre via la Base de Registre où il faut donner les droits en **Contrôle total** à **tout le monde** sur la ruche **26** de la ruche **{eb004a00-xxxxx}** située en **HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Nsi**

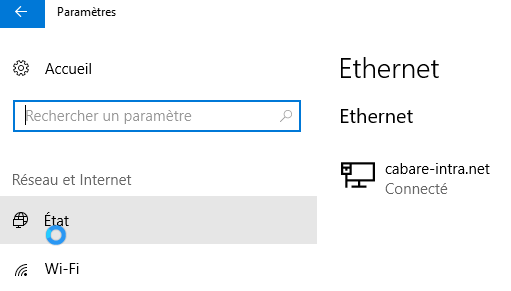


Depuis la version **1607** de windows on peut directement demander depuis l’interface graphique, via les paramètres, une ré-initialisation de TCP-IP. Il faudra impérativement

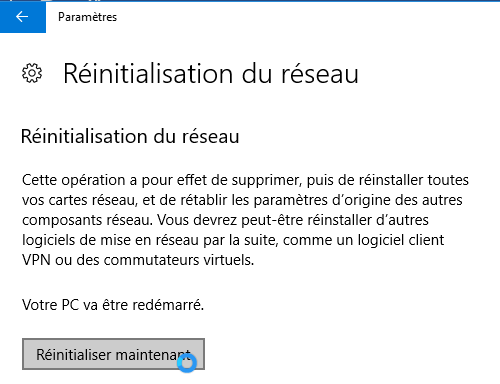
* redémarrer le poste et
* reprendre toutes les configurations réseaux existantes



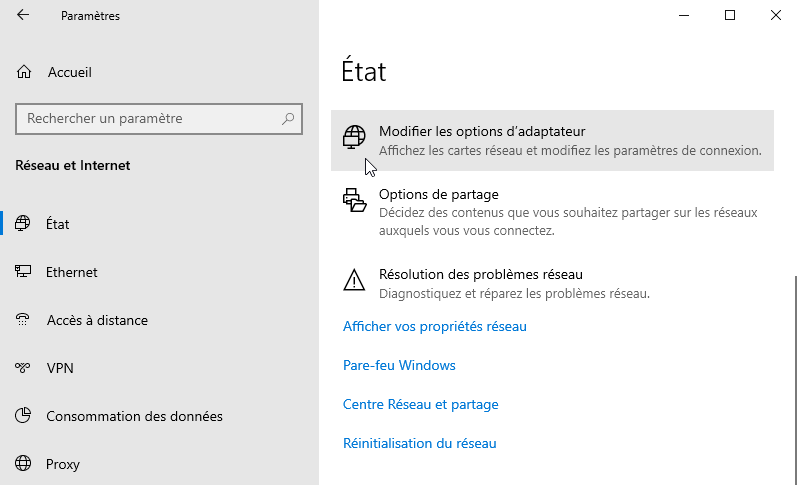
Dans **Etat** on va chercher (tout en bas) **Réinitialisation du réseau**



Et on demande de **Réinitialiser maintenant**



A chaque nouvelle version / builds, des diagnostiques sont ajoutés… en plus



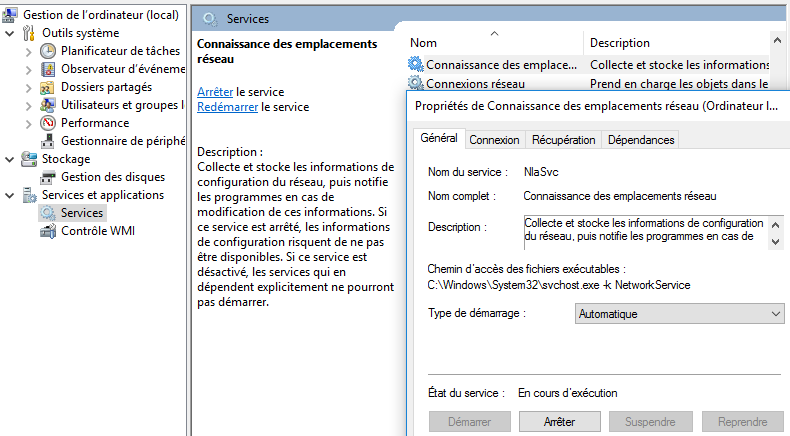
## Profil – Type Réseau Windows 10 :

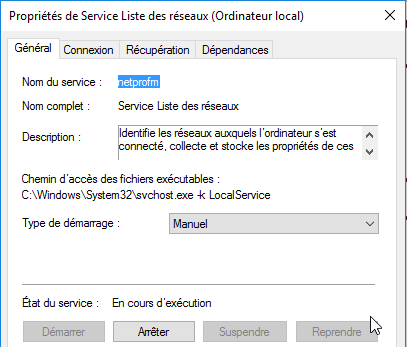
On est avec Windows 10 toujours dans un profil réseau… Dès qu’un réseau est détecté. Il existe trois type de réseau, mais l'utilisateur ne peut choisir que entre **Privé / Public** , car si **Domaine** existe, il ne peut être remis en cause.

* Domaine
* Privé
* Public

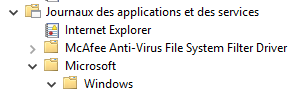
La différence entre **Public / Privé** est une différence des paramétrages par défaut, disponible dans les "partages avancés" et dans le pare-feu.

C'est le service "**Connaissance des emplacements réseau**" (**NlaSvc**) qui gère cela. Vérifier qu'il soit bien démarré sur les clients



Il y a un service dépendant également qui intervient c’est "**Service Liste des réseaux**" (**Netprofm**)

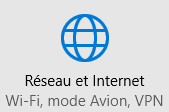
Dans l’observatoire d’évènement les sources: **Microsoft-Windows** avec **NlaSvc** et **NetworkProfile** peuvent aider pour un diagnostic





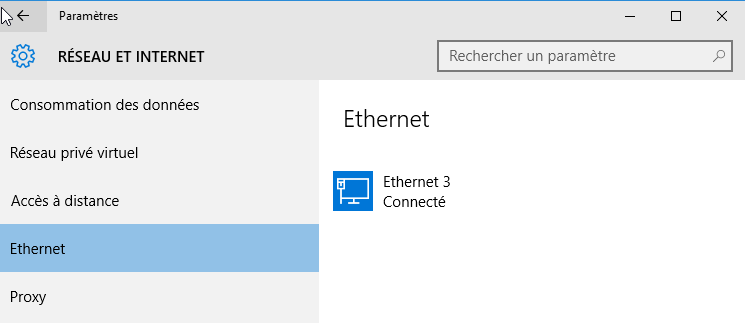
## Changer de type de Profil réseau – interface Paramètre :

Dans l’interface Windows 10 on demande **Paramètres** 

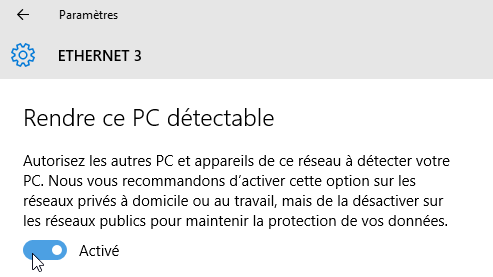
et ensuite **réseau et internet**

On se place ensuite sur **Ethernet,** et on sélectionne la carte réseau pour laquelle on veut modifier le type de Profil

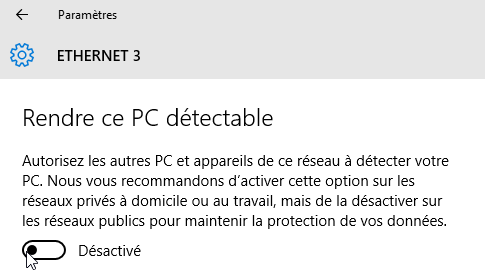
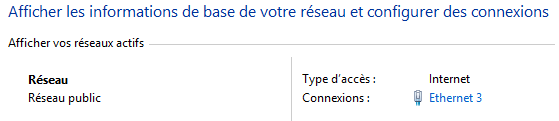
Dans l’exemple **Ethernet 3**



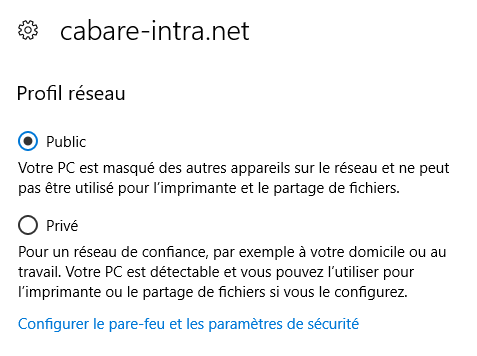
Dans la boite de dialogue qui s’ouvre, on ne choisit pas public ou privé, mais simplement le fait de dire **Rendre ce pc détectable**, fera le réseau de type **Réseau privé**



Et si on indique ne pas vouloir, alors cela fera le réseau de type **Réseau public**

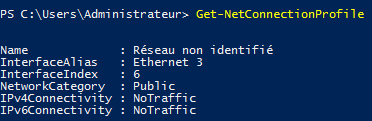


**N.B** : sous la version **Windows 10 - 1709** la boite de dialogue change, et on a de nouveau la mention **profil réseau public / privé** qui apparaît (à la place de detectable…)



## Changer de type de Profil réseau – Powershell :

In faut d’abord récupérer le nom du progfil réseau en cours par la commande • **Get-NetConnectionProfile**



Puis passer une commande du genre **Set-NetConnectionProfile**

Avec 2 paramètres **- name (et entre guillemets le nom du profil)**

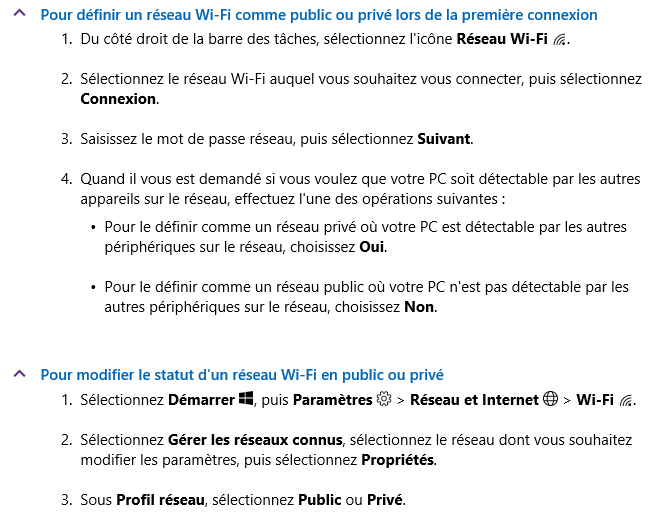
et **- NetworkCategory (avec mot clé Private ou Public)**



## Changer de type de Profil réseau WI FI – windows 1709

Cela n’est possible que si la connexion est uniquement en WIFi

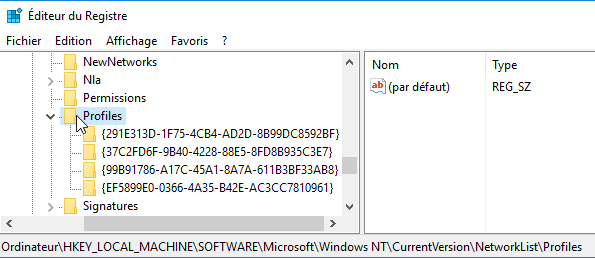
Si une connexion en RJ45 est active en parallèle, on ne peut choisir le type de réseau pour la connexion WIFI



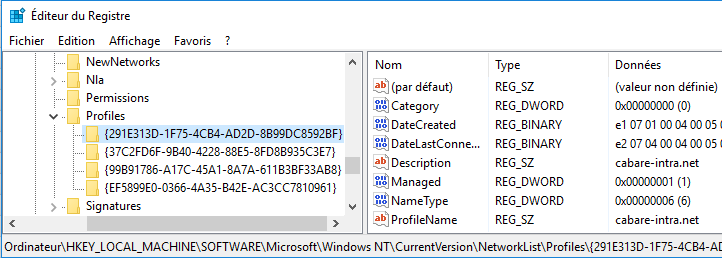
## Changer de type de Profil réseau – Regedit :

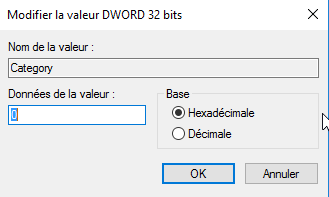
Se placer dans la clé suivante :

**HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Profiles**



Rechercher dans les sous-clé, (ici dans l’exemple il y en a **4**) celle correspondant à votre réseau (la valeur **ProfileName** doit porter le nom de votre connexion réseau, ici dans l’exemple **cabare-intra.net**)



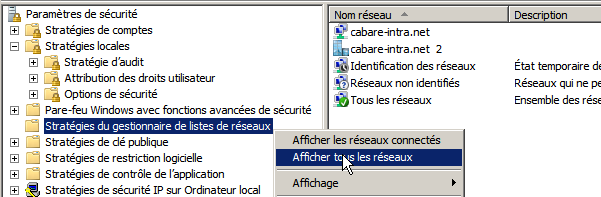


La valeur **DWORD Category** correspond au type de profil (Public/Privé/Domaine) avec les conventions suivantes

Public 0 Privé 1 Domaine 2

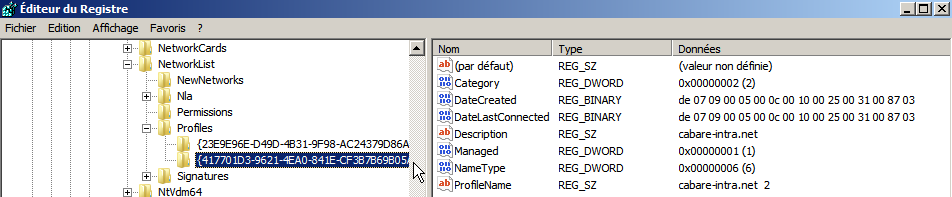
## Reset - Listes des réseaux identifiés

**N.B** : la liste de tous les réseaux détectés par Windows se trouve en demandant dans les **stratégies de sécurité locales**, dans les **stratégie du gestionnaire de liste de réseaux / Afficher tous les réseaux**



**N.B** : la liste des réseaux détectés par Windows est stockée en dans la base de Registre en

**HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Profiles**



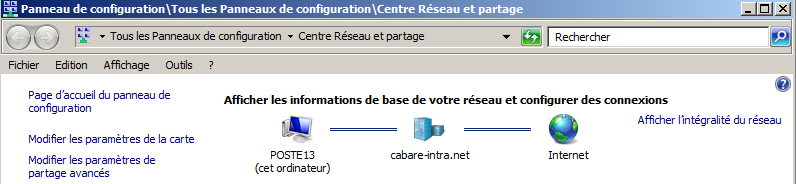
Pour faire un **Reset** de la situation il peut être necéssaire de

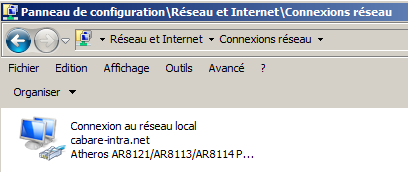
* Désactiver la carte réseau
* Modifier l’adressage IP
* Purger les **NetworkList\Profiles** de la base de registre
* Re-demarrer le service **Service Liste des réseaux (netprofm)**
* Réactiver la carte réseau

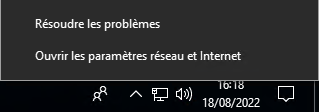
# Réseau Windows 7

## Paramétrage TCP/IP Windows:

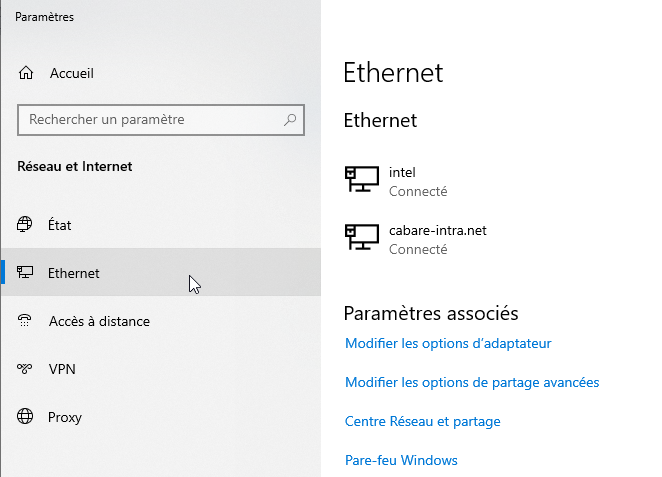
selon les versions , Clic / droit sur l'icône **Réseau** du bureau **,**



On demande **Modifier les paramètres de la carte**, et on choisit notre carte réseau…

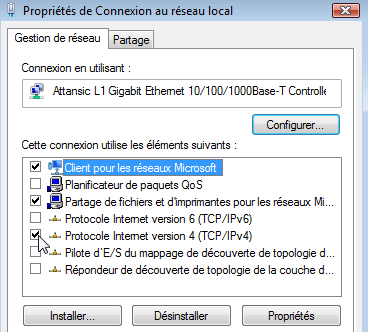
ou de la barre des tâches et via clic/droit demander **Ouvrir les paramètres Réseau et Internet"**

Sur des postes Windows 10, cela amène une autre boite de dialogue, où, dans **Ethernet**, on demande ensuite **Centre Réseau et Partage**



**N.B**: accès possible également via les paramètres de Windows 10 / **Réseaux et internet** / puis on retrouve **Ethernet / Centre Réseau et partage**

pour retrouver enfin les propriétés



Il est important dans un premier temps de :

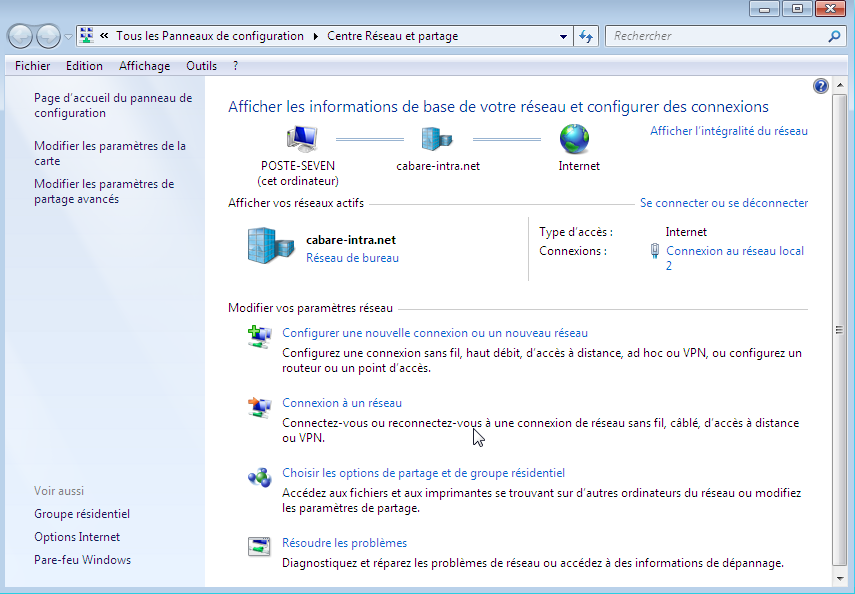
* Désactiver IPv6
* les nouveaux protocoles de découverte Windows...

## Accès au Centre Réseau Windows:

Si une carte réseau (minimum) est installée correctement, une icône "réseau" devrait apparaître en bas à droite… 

Quel que soit son aspect…    

Devrait amener



On peut aller aussi aux réglages du pare-feu

**Modifier les paramètres de partage avancés**

Permet de gérer les paramètres des "profils réseaux". On est depuis windows SEVEN, et donc avec 10 toujours dans un profil réseau…

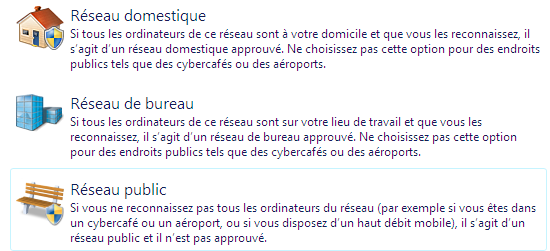
**N.B**: Il existe trois type de "profils" réseau au sens Windows, mais l'utilisateur ne peut choisir que entre **Privé / Public** , car si **Domaine** existe, il ne peut être remis en cause:

**Modifier les paramètres de la carte**

Permet pour chaque carte, de configurer les protocoles, dont TCP-IP…

## Profil – Type Réseau seven 7:

On est avec SEVEN toujours dans un profil réseau… Des que un réseau est détecté (pour ce faire il faut un adressage IP minimum, Adresse/Masque + Passerelle) alors il faut décider de son "type" parmi ceux proposés



De l'appartenance à tel ou tel type de réseau, découleront

Les paramètres des partages avancés

Les réglages du Pare-Feu

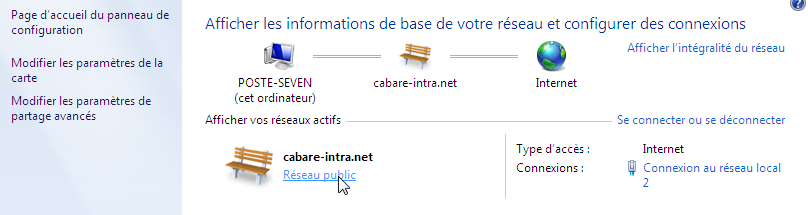
Il existe trois type de réseau au sens SEVEN, mais l'utilisateur ne peut choisir que entre **Privé / Public** , car si **Domaine** existe, il ne peut être remis en cause:

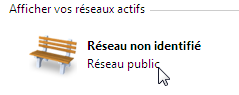
* Domaine
* Privé
* Public

La différence entre **Public / Privé** est une différence des paramétrages par défaut des "partages avancés" et du pare-feu.

## Choisir un profil Réseau 7 :

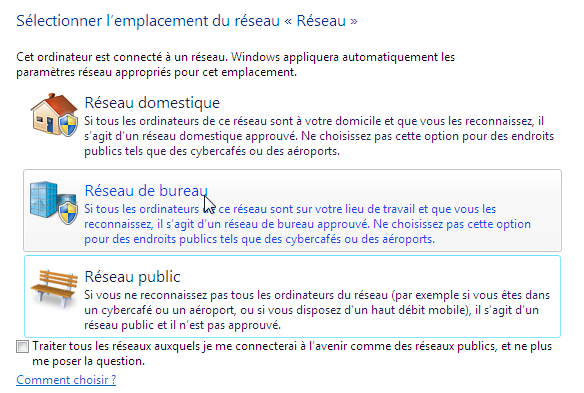
Si on n'est pas dans un domaine, alors on peut en cliquant sur le type de réseau en cours… choisir le nouveau profil réseau.



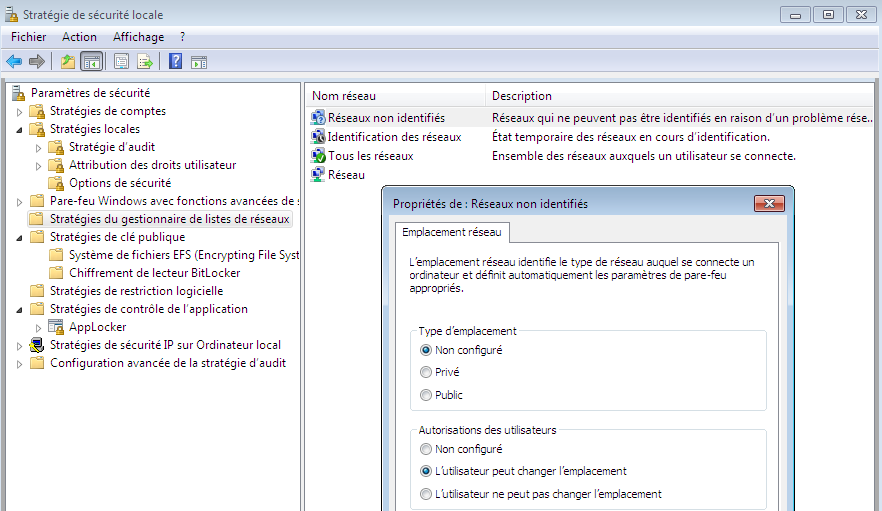
**N.B**: Si **Réseau non identifié** s'affiche,   
une passerelle sera la bienvenue…!

Car dans ce cas on ne peut pas changer de profil…

Des que l'on donne une adresse IP en passerelle, cela débloque la détection, et windows nous demande de choisir un profil



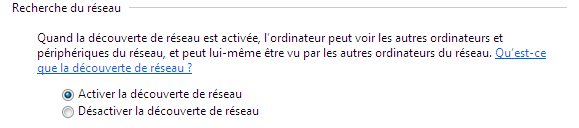
Cet état de fait est réglable dans les **stratégies locales de sécurité**, dans les **Stratégies du gestionnaire de liste de réseaux / Réseaux non identifiés**



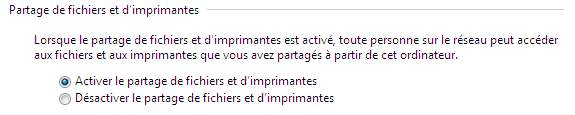
# Profil réseau avancé – Voisinage Réseau

## Réglage Disponibles:

Les réglages suivants existent selon les profils réseaux… Surtout



Pour voir les postes dans les favoris réseaux…



Mais pour que cela soit possible, il faut absolument que les 4 services suivants soient démarrés sur le poste  (ce qui n’est pas toujours le cas):

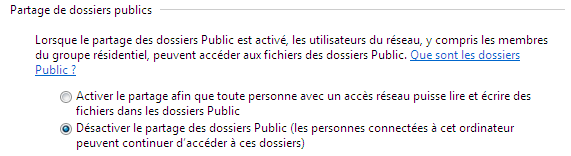
**Client DNS** **Publication des ressources…**

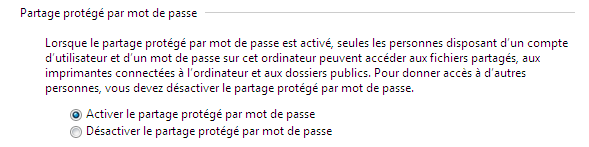


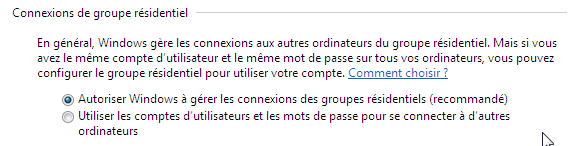
**Découverte SSDP Hôte de périphérique**

**Dossiers publics** et **groupe résidentiels** ne devraient pas être activés sur des machines professionnelles…



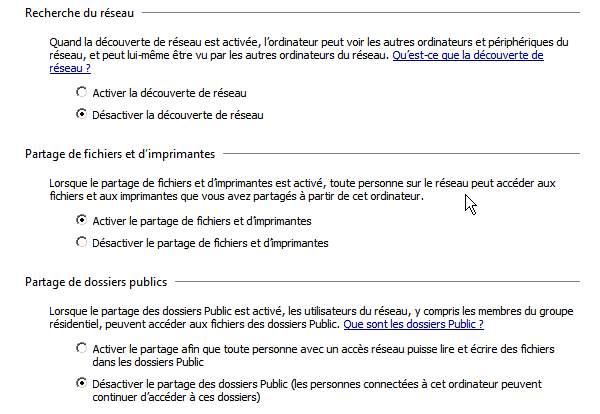




## Jeux de Réglages:

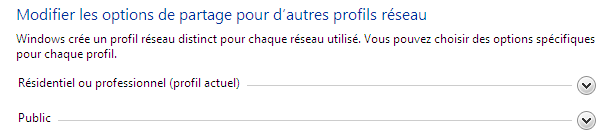
Le jeu des réglages est quasiment le même dans tous les profils...

à chaque emplacement correspond un "jeu de réglage" pré-réglé



Sur une machine en workgroup deux profils type existent

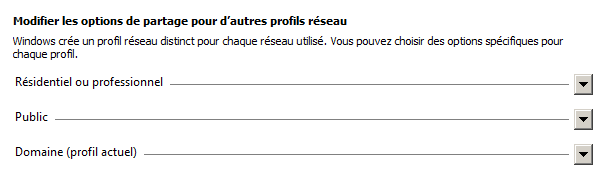
* un répondant au label "**privé**"



* un répondant au label "**public**"

Sur une machine appartenant à un domaine un troisième profil apparaît (et on ne peut pas en choisir un autre)

* répondant au label "**domaine**"



## Activer la découverte du "voisinage réseau":

On a indiqué que 4 services devaient être démarrés sur le poste pour que notre ordinateur puisse être vu, Pour le voisinage réseau il en faut ajouter 2

* Hôte du fournisseur de découverte de fonctions (fdPHost)
* Mappage de découverte de topologie de la couche de liaison (lltdsvc)



Donc au total 6 services

* Client DNS
* Publication des ressources de découverte de fonctions
* Découverte SSDP
* Hôte de périphérique UPnP
* + Hôte du fournisseur de découverte de fonctions
* + Mappage de découverte de topologie de la couche de liaison

**Client DNS** **Publication des ressources…**

Service **Dnscache** Service **Fdrespub**

**Découverte SSDP Hôte de périphérique**

Service **SSDPSRV** Service **upnphost**

**Hôte fournisseur découverte… Mappage de découverte de topologie…**

Service **fdpHost** Service **lltdsvc**



Pourquoi pas un script du genre en Powershell pour les démarrer…

$list = "Dnscache","FDResPub","SSDPSRV","upnphost","fdPHost", "lltdsvc",

foreach ( $service in $list) {

$TheService = get-service | ?{$\_ -like "\*$service\*"}

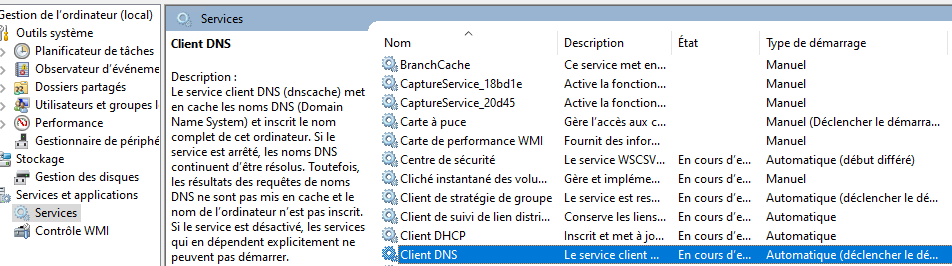
$TheService

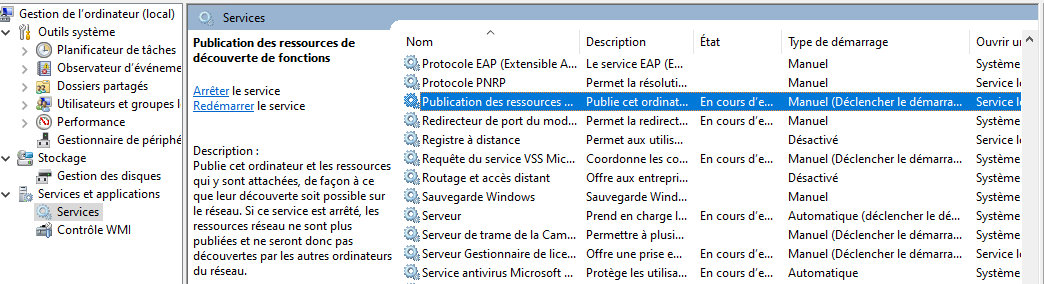
if ( $TheService.Status -eq "Stopped" ){

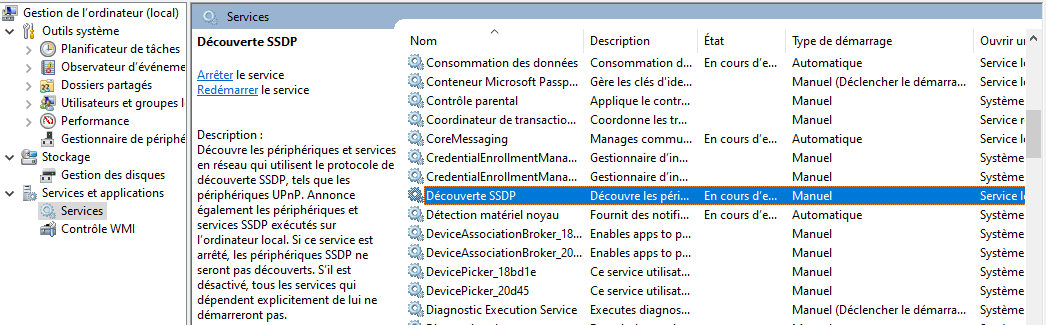
set-service $service -StartupType Manual -Status Running

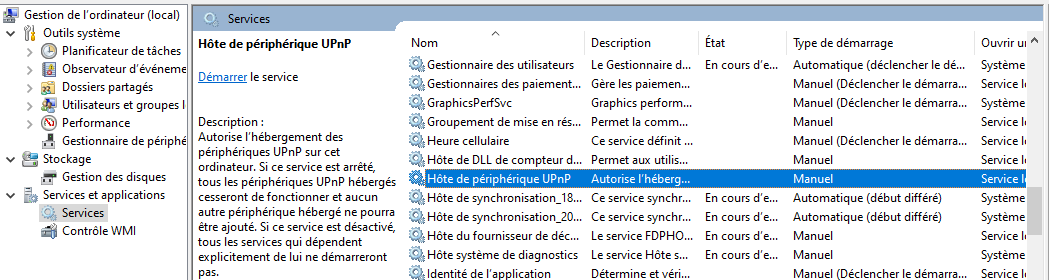
}

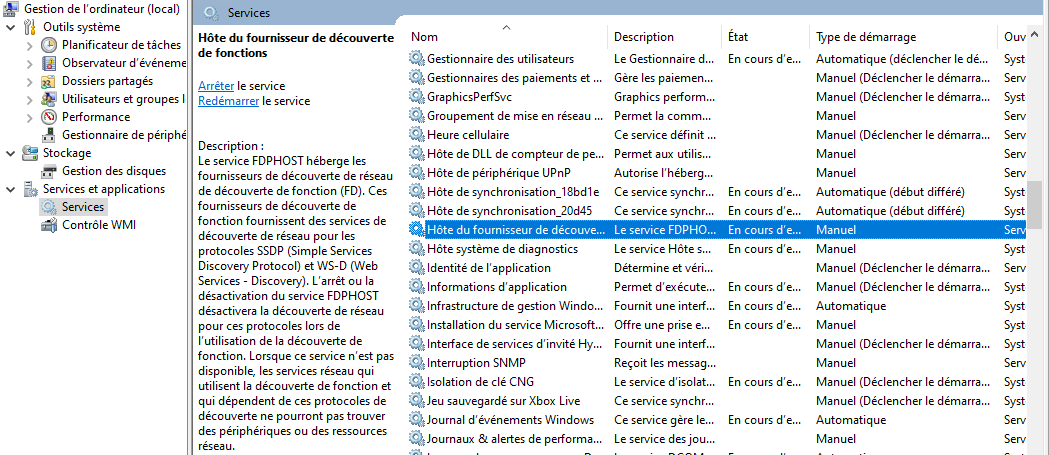
}

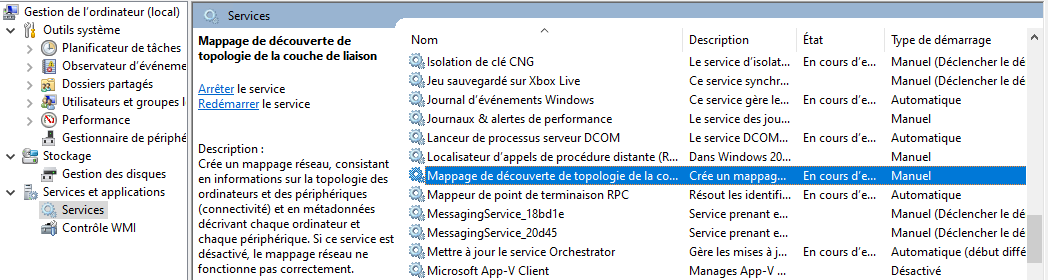












# Mécanisme du Voisinage réseau

## Principe de fonctionnement :

Lorsque l'on clique sur voisinage réseau, on a souvent une réponse lors du démarrage de la machine comme quoi le "parcours du réseau est impossible", or **il suffit d'attendre et tout rentre dans l'ordre**...

Mais la signification du message est la suivante : actuellement un **Explorateur Principal** n'est pas encore identifié...

Environ toutes les 12 minutes, les serveurs annoncent leur présence avec des trâmes spéciales au format NetBios. Une élection d' Explorateur Principal peut arriver lorsque

1. un ordinateur n'arrive pas à trouver un Explorateur Principal
2. Lorsque un Explorateur Principal arrive sur le réseau, ou s'arrête.
3. Lorsque un Contrôleur de Domaine démarre:

Lorsque une élection est lancée, un algorithme compliqué basé sur plusieurs variables se déroule (type de OS, version d'OS, configuration, adressage IP, nombre de machines présentes etc) et un seul Explorateur Principal sera déclaré !

A chaque fois qu'un PC démarre, il est configuré par défaut pour tenter de savoir s'il doit devenir Explorateur...

Il peut exister jusqu'à 5 types de machines dans un réseau Windows

**Non-Browser / Non Explorateur**

Un **non-browser** ou **non Explorateur** est un ordinateur qui a été configuré pour ne pas maintenir une liste des ordinateurs devant apparaître dans le voisinage réseau

**Potential Browser / Explorateur Potentiel**

Un **Potential-Browser** ou **Explorateur Potentiel** est un ordinateur capable de maintenir une liste des ordinateurs devant apparaître dans le voisinage réseau , et pouvant être promu comme Explorateur principal. Un **Explorateur Potentiel** est aussi capable de jouer le rôle d'un **Explorateur de Secours**, s'il est piloté par un **Explorateur Principal**

**Backup Browser / Explorateur de Secours**

Un **Backup-Browser** ou **Explorateur de Secours** reçoit une copie des ordinateurs devant apparaître dans le voisinage réseau depuis un **Explorateur Principal** et fournis cette liste à la demande des autres ordinateurs du domaine ou du groupe de travail

**N.B:** Lorsqu'un poste démarre, c'est l' **Explorateur Principal** qui lui indique s'il doit devenir un **Explorateur de Secours** ou non

**Master Browser / Explorateur Principal**

Un **Master-Browser** ou **Explorateur Principal** est responsable de la collecte des informations nécessaires à la création et à mise à jour de la liste des ordinateurs figurant dans le voisinage réseau. Cette liste inclus tous les serveurs du domaine de l' **Explorateur Principal** et la liste de tous les domaines sur le réseau. Les machines windows annoncent leur présence à l' **Explorateur Principal** par un datagrame appelé "server annoucement", et celui-ci les ajoute

* Si un Domaine s'étends sur plus d'un sous-réseau, l' **Explorateur Principal** travaille de la manière suivante :
* Il gère la liste pour le sous-réseau dont il fait partie
* fournit cette liste à chaque Explorateur de Secours de chaque sous-réseau
* Si un sous-réseau comprends plusieurs Domaines, chaque Domaine à son  **Explorateur Principal** et éventuellement ses **Explorateurs de Secours**

**Domain Master Browser / Explorateur Principal de Domaine**

Un **Domain Master-Browser** ou **Explorateur Principal de Domaine** est responsable de la collecte des informations pourla création et la mise à jour de la liste pour tout le domaine, collecte les informations des **Explorateur Principaux** des autres sous-réseaux et fournit les informations aux **Explorateur Principaux** des autres sous-réseaux.

Un **Explorateur Principal de Domaine** est toujours le Contrôleur Principal de Domaine

**N.B:** Un poste peu jouer plusieurs rôles, par exemple l' **Explorateur Principal** peut aussi être un **Explorateur Principal de Domaine**

## Rafraîchissement Tests et vérifications :

Quelles sont les vitesses de raffraîchissement ?

de quelques secondes, à plusieurs minutes, jusqu'à 12 minute pour la prise en compte d'un serveur dans un Domaine, ce qui par rebonds peut aller à 24 minutes entre 2 Domaines...

Pour la suppression d'une machine c'est pire, Microsoft annonçant jusqu'à 36 minutes pour la mise à jour d'une liste "rayant" une machine qui ne se serait pas correctement déconnectée du réseau (arrêt système brutal...)

## Peut on éviter l'élection d'un Explorateur ? :

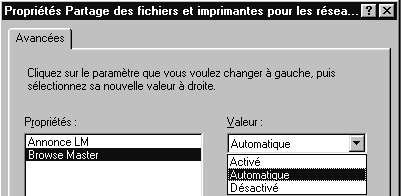
la réponse est non; il doit y en avoir toujours un, mais on peut a la limite accélérer un peut les choses

**En implémentant un serveur WINS** qui diminuera le trafic réseau pour les résolutions de nom Netbios,

**En implémentant un serveur DNS** qui diminuera le trafic réseau pour les résolutions de nom

**En modifiant le status d'une machine** : si on modifie dans propriété de partage des fichiers et imprimantes le fait qu'une machine soit éligible ou non (on peut éviter les élections et diminuer les trâmes émises...)

Sous Windows 95-98

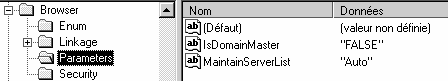


définir qui peut être Browse Master

**N.B**: Il doit y en avoir toujours 1 seul !

Sous Windows NT ou 2000

Il faut modifier la base de registre NT "ce qui reste délicat"

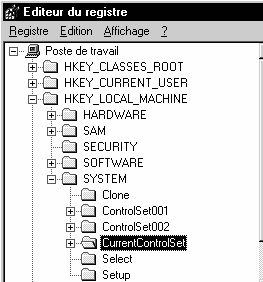


Il faut se positionner sur la clé

**HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\Browser\  
Parameters**

et y modifier la clé de type DWORD-value nommée **MaintainServer List**les valeurs possibles sont **"Auto" "No" et "Yes"**

**En accélérant la vitesse de rafraîchissement**...Il faut modifier la base de registre NT "ce qui reste délicat"



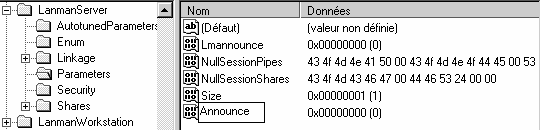
Il faut se positionner sur la clé **HKEY\_LOCAL\_MACHINE\SYSTEM\  
CurrentControlSet\Services\LanmanServer\  
Parameters**

et y créer une clé de type DWORD-value

en allant dans le menu

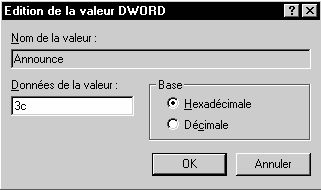
**Edition / nouveau / valeur Dword**

et y entrer la clé **Announce**



cette valeur Announce il faut ensuite la modifier via le menu

**Edition / modifier**



une valeur de 60 secondes (3c hexa) semble un bon compromis entre vitesse et nombre de trâmes...

# Protocole DHCP

## Objectif de DHCP :

Le protocole **DHCP** (**Dynamic Host Configuration Protocol**) centralise et gère l’attribution des informations de configuration **TCP-IP** en affectant automatiquement des adresses **IP** à des ordinateurs configurés pour utiliser DHCP. La mise en œuvre de **DHCP** élimine certains problèmes de configuration liés à la configuration manuelle de **TCP-IP**.

A chaque démarrage d’un client **DHCP**, ce dernier demande des informations d’adressage IP à un serveur **DHCP**. Un client ne choisit pas un serveur DHCP, il interroge le réseau avec un **broadcast DHCP** pour repérer les serveurs **DHCP** potentiel en vue de récupérer a terme notamment :

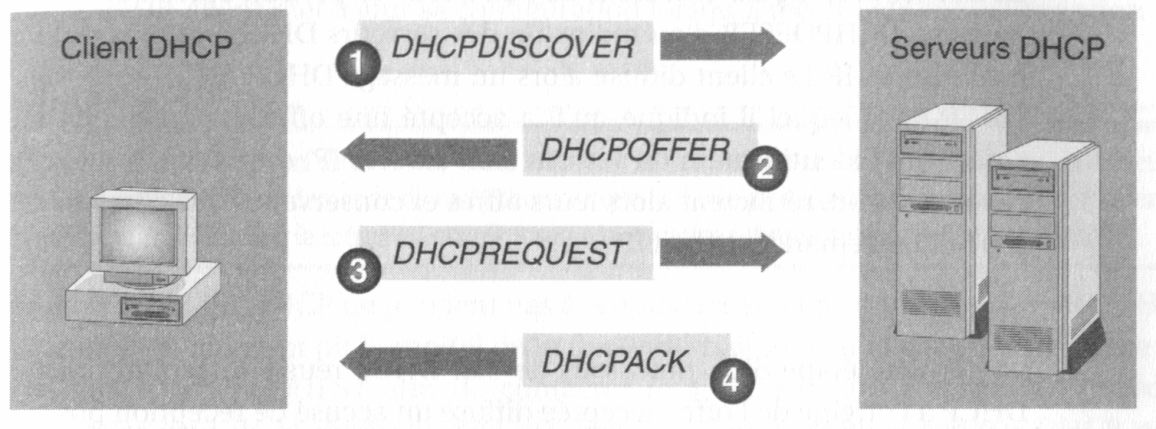
* Une adresse IP
* Un masque de sous-réseau.
* Des valeurs facultatives, comme une adresse de **passerelle** par défaut, une adresse **DNS** ou l’adresse du serveur WINS.

Lorsqu’un serveur **DHCP** reçoit une requête, il sélectionne des informations d’adressage IP dans une réserve d’adresses définie dans une base de données et les propose au client **DHCP**. Si le client les accepte, les informations d’adressage **IP** lui sont cédées sous la forme d’un bail d’une durée spécifique. Si aucune information d’adressage IP n’est disponible dans la réserve pour location au client, ce dernier ne peut pas initialiser **TCP/IP**. Il peut selon les cas se doter d'une adresse **APIPA**.( cf chap spécifique).

**Remarque** : Le protocole **DHCP** est défini dans les RFC 1533, 1534, 1541 et 1542. et est dérivé du protocole **BootP**.

## Fonctionnement de DHCP :

Pour configurer un client DHCP, le protocole DHCP travaille en 4 phases :



### DHCPDISCOVER ou "Demande de bail IP" :

Le client ne disposant pas d’adresse IP et ne connaissant l’adresse IP d’aucun serveur, il utilise 0.0.0.0 comme adresse de source et 255.255.255.255 comme adresse de destination.

La demande de bail est envoyé au sein d’un message **DHCPDISCOVER**. Ce message contient également l’adresse matérielle et le nom d’ordinateur du client, afin que les serveurs DHCP puissent identifier l’émetteur de la requête. Tous les serveurs répondent s'ils le peuvent.

Le processus de bail IP est utilisé dans l’une des situations suivantes:

* TCP/IP est initialisé pour la première fois en tant que client DHCP.
* Le client demande une adresse IP spécifique qui lui est refusée. Il est possible que le serveur DHCP ait supprimé le bail.
* Le client disposait auparavant d’un bail d’adresse IP mais y a mis fin et en demande un nouveau.

### DHCPOFFER ou "Offre de bail IP" :

Tous les serveurs DHCP qui ont reçu la demande et qui disposent d’une configuration valide vis-à-vis du client diffusent une proposition.

Le client ne disposant pas encore d’une adresse IP, l’envoi de la proposition s’effectue par diffusion sous forme de message **DHCPOFFER.**

**Remarque :** Lorsque aucun serveur DHCP n’est en ligne, le client DHCP attend une proposition pendant 1 seconde. S’il n’en reçoit aucune, il diffuse à nouveau la requête à trois reprises (selon des intervalles successifs de 9, 13 et 16 secondes). Si aucune proposition n’est reçue après quatre tentatives, le client essaie à nouveau toutes les 5 minutes.

### DHCPREQUEST ou "Selection de bail IP" :

Après avoir reçu une proposition d’au moins un serveur DHCP, le client informe par diffusion tous les autres serveur DHCP de sa sélection, en acceptant la première proposition reçue.

La diffusion est envoyé dans un message **DHCPREQUEST** et comprend l’identificateur du serveur (AI) dont la proposition a été acceptée. Tous les autres serveurs DHCP retirent leur proposition afin que les adresses IP dont ils disposent restent disponibles pour la requête de bail IP suivante.

### DHCPACK / NACK ou "Accusé de réception de bail IP" :

Le serveur DHCP dont la proposition est acceptée diffuse au client un accusé de réception stipulant la conclusion du bail, sous la forme d’un message **DHCPACK**. Ce message contient un bail valide pour une adresse IP et éventuellement d’autres informations de configurations.

Si un accusé de réception stipulant la non conclusion du bail (**DHCPNACK**) est diffusé (le client tente de souscrire le bail d’une adresse IP dont il disposait précédemment alors que cette adresse n’est plus disponible par exemple) le client retourne au processus de demande de bail IP.

### "Renouvellement de bail IP" :

Tous le clients DHCP tentent de renouveler leur bail lorsqu’il atteint **50 %** de sa durée. Pour renouveler, un client DHCP envoie un message **DHCPREQUEST** directement au serveur DHCP avec qui il a conclu le bail en vigueur.

Si le serveur DHCP est disponible, il renouvelle le bail et envoie au client un accusé de réception stipulant la conclusion du renouvellement (**DHCPACK**) et la nouvelle durée, ainsi que les éventuelles mises à jour des paramètres de configuration.

Lorsque le client reçoit l’accusé de réception, il met à jour sa configuration. Si un client tente de renouveler son bail mais est dans l’impossibilité de contacter le serveur DHCP à l’origine de ce dernier, le client peut encore utiliser l’adresse, puisqu’il lui reste 50 % de la durée du bail.

Lorsqu’un client DHCP redémarre, il tente d’obtenir un bail pour la même adresse avec le serveur DHCP d’origine. Pour ce faire, il diffuse un message **DHCPREQUEST** spécifiant la dernière adresse IP dont il avait le bail. Si la tentative se solde par un échec et qu’il lui reste encore du temps avant l’expiration du bail, le client DHCP continue à utiliser la même adresse IP.

Si un bail, lorsqu’il atteint **50 %** de sa durée, n’a pas pu être renouvelé par le serveur DHCP d’origine, le client tente de contacter les autres serveurs DHCP disponibles lorsque **87,5% du temps s’est écoulé**. Le client diffuse alors un message **DHCPREQUEST**. Tous les serveurs DHCP peuvent répondre par un message **DHCPACK(renouvellement du bail)** ou **DHPCNACK (obligeant le client DHCP à se réinitialiser** et à obtenir le bail d’une adresse IP différente).

Lorsque le bail expire ou qu’un message DHCPNACK est reçu, le client DHCP doit immédiatement cesser d’utiliser l’adresse IP. Il retourne alors au processus de souscription d’un nouveau bail d’adresse IP.

### DHCPRELEASE ou libération des ressources:

Le client peut envoyer un message DHCPRELEASE lorsqu'il s'arrête. Ainsi le serveur DHCP peut de nouveau utiliser ces adresses pour un autre client...

**N.B:** Microsoft n'utilise pas cette commande. Lorsqu'une machine s'arrête, son Bail courre encore sur le serveur DHCP. Si le client se reconnecte au réseau avant la fin du bail, son bail sera réattribué par un demande DHCPREQUEST...

# Client DHCP

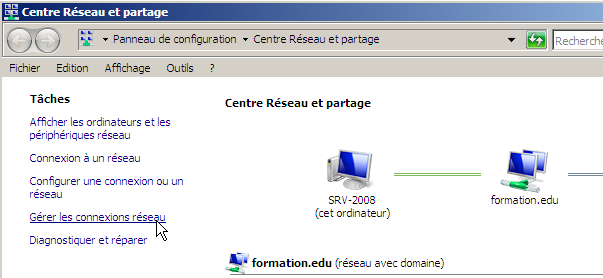
## Client DHCP Windows 10 - Seven

Un poste devient client DHCP simplement en demandant dans le paramétrage de TCP/IP « **Obtenir automatiquement une adresse IP** »

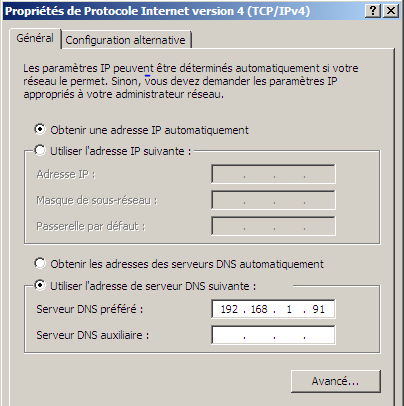
soit par **propriétés** de **réseau**,(sur le bureau)

**Gérer les connexions réseau**

soit par **démarrer / paramètres / panneau de configuration / centre réseau et partage / Gérer les connexions réseau**



puis **propriétés** de **Protocole Internet Version4** (**TCP/IPv4**)



Attention, on ne gère pas forcément l’adresse Ip est le paramétrage DNS de la même manière…

**N.B**: Lorsque l'on demande une **adresse automatique**, tout le reste du paramétrage IP devient "inactif"

## Ipconfig /release /renew :

Depuis **Seven** (et à partir de NT) , à travers l'utilitaire **ipconfig** on peut demander de libérer – renouveler une adresse reçue dynamiquement...par les options

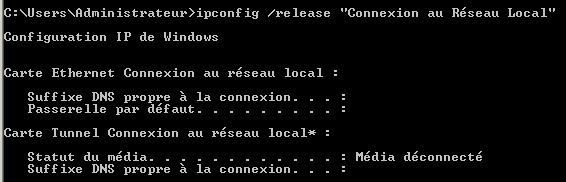
**Ipconfig /release** et

**Ipconfig /renew ...**

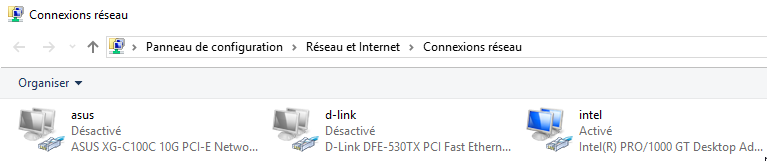


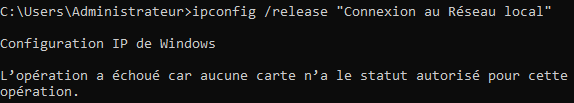
Depuis **Seven** si plusieurs cartes existent, (et plusieurs protocole) on peut cibler le périphérique de destination de la commande **ipconfig**…

On parle du nom logique de la carte et nom du nom technique. Si la carte est nommée par défaut, cela sera "***Connexion au Réseau local***"



Mais si les cartes ont été nommées…





**N.B:** si aucun serveur DHCP n'est présent, un mécanisme dit "adresses APIPA" se met en œuvre, (voir "adresses automatiques APIPA") uniquement pour des postes **Windows 10,** **Seven, (Windows )**

**N.B:** attention à la possibilité d’une **Configuration alternative**….(voir chapitre adresse **APIPA**)

# Adresses Ip automatiques (APIPA)

## Principe APIPA et DHCP:

L’origine du mécanisme vise à pallier une défaillance du Serveur DHCP.

Le fonctionnement est le suivant :

1. Une machine installée avec un protocole TCP/IP tente de contacter un serveur DHCP pour recevoir une adresse IP de manière dynamique (elle doit être configurée pour...)
2. Si aucun serveur DHCP ne répond, la fonction APIPA génère une adresse IP au format 169.254.xxx.xxx avec un masque de sous-réseau 255.255.0.0. Si cette adresse est déjà utilisée la fonction APIPA en sélectionne une autre pour un maximum de 10 coups.
3. Une fois une adresse prise, l'ordinateur la diffuse et l'utilise jusqu'à ce qu'un serveur DHCP n'apparaisse opérationnel sur le réseau !

Quelques remarques :

* l'IANA (Internet Assigned Number Authority) à réservé les adresses de **169.254.0.0** à **169.254.255.255** à la fonction APIPA, ces adresse n'étant pas routables !
* Les machines utilisant des adresses APIPA ne peuvent communiquer qu'avec des machines faisant partie du même sous-réseau de classe B, et dotée d'un adresse au format 169.254.xxx.xxx

## APIPA et Windows:

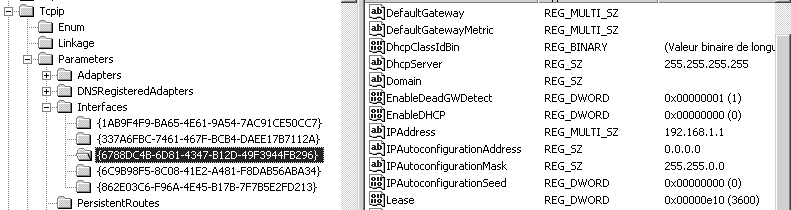
Pour que Windows puisse gérer les adresses APIPA, il est nécessaire d'utiliser TCP/IP comme protocole et de demander le bouton Option "Obtenir une adresse IP automatiquement" dans Propriétés de Protocole Internet (TCP/IP). Il s’agit en fait de configurer le client DHCP.

## Désactivation adresse APIPA:

Par défaut les adresses APIPA sont actives, il est possible de les inhiber en allant dans la base de registre et en demandant

Pour chaque carte réseau sélectivement :

**HKEY\_LOCALMACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\GUID\_carte\_réseau** et en lui ajoutant l'entrée

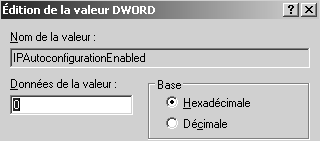
****

**IPAutoconfigurationEnabled** avec une valeur de 0

(si cette entrée n'existe pas ou que sa valeur est fixée à 1 APIPA est activée)

On peut aussi invalider les adresses **APIPA** globalement pour toutes les cartes en ajoutant la même clé

**IPAutoconfigurationEnabled** avec une valeur de 0



Directement au niveau de l’entrée

**…CurrentControlSet\Services\Tcpip\Parameters**

## Adresse IP alternative:

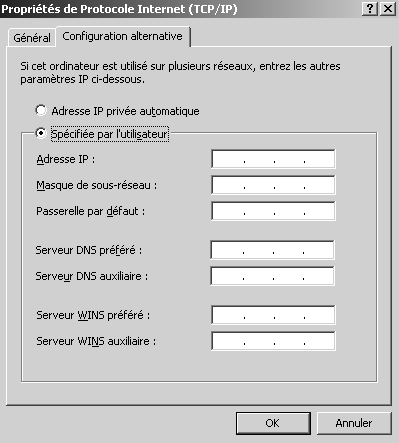
Si un client DHCP ne trouve pas de serveur DHCP, il peut donc prendre une adresse APIPA.

Mais il est possible de lui spécifier une adresse alternative, qui lui sera attribuée dans le cas ou un serveur DHCP est manquant. Et donc prenant la place du mécanisme APIPA.

Cela peut permettre ainsi de pouvoir avoir sur un portable, une configuration « Bureau » en tant que client DHCP, et une configuration « maison » avec une adresse privées classique.

**N.B :** seuls les **Admins** ou **Opérateurs de configuration Réseau** peuvent modifier ce paramétrage.

Lorsque sur une carte on est en client DHCP, alors un onglet supplémentaire est activé : l’onglet **Configuration alternative**



Il est possible ici d’indiquer une configuration complète…

**N.B :** si on utilise ce mécanisme de **Configuration Alternative**, il ne faut pas alors dévalider les adresses APIPA avec la modification de la base de registre du chapitre précedent.

Toute présence de clé **IPAutoconfigurationEnabled** annulera ce mécanisme

# Notion de DNS

## Le DNS:

DNS est au centre de la gestion des domaines dans Windows. Il faut en comprendre certaines notions fondamentales

### Noms DNS

Selon la définition de la RFC 952 le nom DNS d'un ordinateur est constitué de plusieurs parties séparées par des virgules, par exemple, **www.fnac.presse.fr.**

|  |  |  |
| --- | --- | --- |
|  | NetBIOS | Full computer name |
| Type | Flat | Hierarchical |
| Character Restrictions | A-Z, a-z, 0-9, "espace",  symbols: ! @ # $ % ^ & ' ) ( . - \_ { } ~ Unicode chars, | A-Z, a-z, 0-9,  symbols: -\_,  Unicode chars.  Le point '.' est le séparateur |
| Maximum Length | 16 (dont 1 réservé) dont 15 en pratique | 63 pour un nom de domaine  255 pour un FQDN |
| Name Service | NBNS (WINS and broadcast) | DNS |

Windows utilise les noms DNS comportant des caractères soulignés, une fonction qui affectera le choix de serveur DNS

### Nom "Plat" Netbios

Les nom netbios sont crées-enregistrés lors du démarrage de chaque poste, et doivent être uniques sur tous le réseau. Ce simple constat pose les limites d'envergure des noms Netbios gérés par broadcast, d'ou l'apparition de serveur WINS sur les réseaux de taille moyenne-grande. Mais même ainsi, il parait impossible d'assurer l'unicité sur des réseau de grandes envergure...

### Nom "Hierarchique" DNS

Un nom se termine par un point **.** qu'il n'est plus necessaire d'écrire, mais qui correspond au domaine "racine"

une URL se lit de droite à gauche

**ftp://ftp.univ-lille1.fr(.)**

**Nom de Domaine**:   
peut aussi contenir un sous-domaine

**Ftp**: Protocole transmission

**Ftp** :   
File Transfer Protocol

**http://www.sncf.fr**

**Nom de Domaine**:   
peut aussi contenir un sous-domaine

**http** : Protocole transmission

**www** :   
World Wide Web

Un nom de domaine (**sncf.fr**) se décompose en

* Un Top Level Domain (exemple : **fr**)
* Un nom d'organisation (appelé aussi nom de domaine) (ex : **sncf**)

### Structure des domaines – délégation de zones

Sur un espace de nom, une délégation de zone, signifie que l'on a autorité pour ce domaine.

* L'IANA a autorité pour les domaines de Premier niveau, les organismes du 1° niveau, ont autorité jusqu'au second niveau…
* Les entreprises / particuliers n'ont autorité qu'a partir des autres niveaux.

Les Top Level Domain les plus courants sont:

|  |  |
| --- | --- |
| **Clé** | **Contenu** |
| .com | Entreprise commerciale |
| .edu | éducation |
| .gov | organismes gouvernementaux |
| .mil | organisations militaires |
| .net | intervenant d'internet |
| .org | instance gouvernementale ou institution administrative |

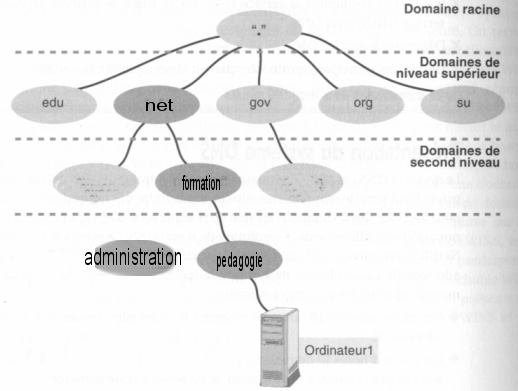
Cependant si ces domaines sont a priori internationaux, ils sont à forte dominante américaine. De plus chaque pays possède son nom de domaine (à l'exception des USA qui utilisent les 6 domaines précédents).

|  |  |
| --- | --- |
| **Clé** | **Contenu** |
| .au | Australie |
| .ca | Canada |
| .fr | France |
| .uk | United Kingdom |

L'internic se chargeant de l'attribution des adresses dans les domaines internationaux, c'est l’AFNIC France qui se charge des attributions des noms de domaine en .fr  **http://www.afnic.fr**

## Zones DNS:

Une **Zone** représente une partie de l'espace de nom de Domaine, à des fins de gestion.



Supposons que vous ayez deux régions, **administration** et **pedagogie**. Chaque région souhaite exploiter un serveur **DNS local**.

Pour répondre aux besoins des deux régions, vous pouvez ajouter un niveau comme par exemple :

**administration.formation.net** et

**pedagogie.formation.net**.

Chaque serveur DNS a une sous-section de domaine (une **zone** en jargon DNS).

Le serveur DNS central **formation.net** ne gère alors plus qu'un très petit nombre de noms de hosts. Il stocke en outre les noms et adresses IP des serveurs DNS de ces zones, à savoir **pedagogie.formation.net** et **administration.formation.net**.

Ainsi, si une machine **ordinateur1** se trouve dans la région **pedagogie**, elle se nommera.**ordinateur1**.**pedagogie.formation.net**

* Si cette machine **ordinateur1** essaye d'atteindre un autre poste du domaine pédagogie, sa requête sera traitée par le serveur DNS de **pedagogie.formation.net**
* Si cette machine **ordinateur1** essaye d'atteindre un poste du domaine administration, sa requête sera traitée par le serveur DNS de **pedagogie.formation.net,** et **redirigée** vers le serveur racine de niveau supérieur**,** à savoir **formation.net.** celui-ci connait le serveur qui gère la zone administration, c'est **administration.formation.net** il renvoit l'adresse de ce serveur DNS au serveur DNS **pedagogie.formation.net** qui peut alors refaire sa demande...

### Zone principale – secondaire

Le serveur DNS peut remplir plusieurs fonctions par rapport à une zone, le serveur chargé de la gestion initiale de la zone est appelé **serveur principal** ou **primary**. mais les informations d'une zone peuvent être répliquées sur d'autres serveurs soit dans un objectif de fiabilité, soit pour un objectif de répartition de charge. Dans ce cas le serveur DNS qui recopie les information depuis le serveur DNS principal s'appelle un **serveur secondaire** ou **backup**. L'édition du fichier de la zone est faite sur le serveur principal qui envoie la version la plus récente du fichier au serveur DNS secondaire. Lorsqu'une machine envoie une requête au serveur secondaire, ce dernier y répond avec sa copie du fichier. Le fichier de zone du serveur secondaire a généralement une durée de vie (généralement de 24 heures). Si le serveur DNS primaire ne met pas à jour le fichier avant la période d'expiration, le serveur secondaire considère l'information comme dépassée. Si votre serveur DNS principal tombe en panne pendant quelques heures, vous n'aurez donc pas de problème. Les serveurs DNS secondaires peuvent être aussi nombreux que l'on le souhaite.

### Requêtes itératives ou récursives

Avec un raisonnement identique à celui précédant pour formation-administration décomposons la requête envoyée à un DNS pour un accès à un site sur Internet.

Vous êtes sur un poste et vous essayez d'atteindre l'URL **www.cabare.net.**

Vous pouvez vous permettre de demander en fait **www.cabare.net**, et cette demande est transmise au serveur DNS de votre FAI.

➐ Qui demande alors a ce DNS l'adresse ip de la machine cabare.net...

➊ www.cabare.net ?

➏ Alors on renvoit l'@ip de ce serveur dns au FAI

➋ Serveur DNS du FAI auquel vous accédez

➍ Sinon renvoit la requête aux serveurs inférieur racine (une dizaine)

Recherche par zone...

➎ Un serveur DNS gère la zone cabare.net

Cela peut être le DNS de l'hebergeur ou le DNS de l'entreprise ou on a déposé le nom...

➌ Si adresse en cache

Alors on renvoit @ip...

* Le processus ➊➋ puis ➌ est appelé requête récursive
* Le processus ➊➋puis ➍➎➏➐ est appelé requête itérative

### Résolution de Noms et Résolution inverse

Chaque composant informatique d'Internet a une adresse IP unique sur 32 bit (par exemple **154.23.17.8**). Il est possible de nommer un élément en se référant à son adresse IP. Mais la plupart des utilisateurs préfèrent les noms plus faciles à retenir comme **http://toto.com**. Pour pouvoir utiliser ce type de noms, il faut une base de données capable de convertir les adresses IP en adresses mémorisables. On appelle cela la **résolution de noms**.

la **résolution de nom** (**forward lookup**) permet de trouver une adresse IP à partir d'un nom

la **résolution inverse** (**reverse lookup**) permet de trouver un nom à partir d'une adresse IP

Du fait du faible nombre de systèmes présents sur Internet à ses origines, les machines connectées à Internet prenaient en charge la résolution de noms via une simple table ASCII (**fichier HOSTS**) qui listait les adresses IP et les noms de machines correspondants. (Le code de TCP/IP permet toujours de placer un fichier HOSTS sur un système). Depuis 1984, les systèmes ont recours principalement à **DNS** pour la résolution de noms. Sinon il faudrait maintenir un fichier HOSTS qui contiendrait non seulement des centaines de millions d'ordinateurs, mais qui changerait quotidiennement !

## Ordre de Résolution DNS par le client Windows :

RECHERCHE HOTE DNS

1. d’abords le cache DNS local en RAM est utilisé
2. Ensuite un fichier Host peut être utilisé
3. le serveur DNS est interrogé (rappel de l’ordre de résolution sur un serveur DNS :)
   1. cache serveur
   2. zone faisant autorité (ou zone déléguée ou zone de stub)
   3. re-directeurs conditionnels
   4. re-directeurs par défaut
   5. indications de racine
4. on enchaîne sur une résolution NetBIOS si le nom est NON FQDN, c'est-à-dire du genre « poste1 » (si le nom est du genre « poste1.domaine.com » alors on n’enchaîne pas sur une recherche NetBios…)
   1. cache local Netbios
   2. serveur WINS
   3. Diffusion Broadcast
   4. Consultation fichier LMHost

**N.B :** il est facile d’effacer le contenu du cache DNS local, par la commande **ipconfig /flushdns**

**Caractéristiques des Serveurs DNS**

L'implémentation la plus populaire de DNS est **BIND** (Berkeley Internet Name Domain) sous UNIX

**DDNS**

La méthode utilisée pour ajouter un nouvel enregistrement correspondant à un nouvel ordinateur - un nouveau host en terminologie DNS, dépend de votre logiciel serveur DNS. La plupart utilisent des fichiers ASCII.

Les solutions de serveur DNS les plus récentes n'exigent plus de mises à jour grâce au standard **DDNS** (**Dynamic DNS**) que décrit en détail la RFC 2136. Dans un réseau compatible DDNS, les ordinateurs font d'eux-mêmes les présentations sans qu'un administrateur ne doive intervenir sur le DNS

**Enregistrements SRV**

Les solutions de serveur DNS les plus récentes gèrent une autre sorte d'enregistrement DNS : les **enregistrements SRV** que décrit en détail la RFC 2052. Ces enregistrements permettent de demander à un serveur DNS si il connaît des machines jouant le rôle de serveur d'un type spécifique

**Serveur principal - secondaire**

Le serveur DNS peut remplir plusieurs fonctions par rapport à une zone, le serveur chargé de la gestion initiale de la zone est appelé **serveur principal** ou **primary**. mais les informations d'une zone peuvent être répliquées sur d'autres serveurs soit dans un objectif de fiabilité, soit pour un objectif de répartition de charge. Dans ce cas le serveur DNS qui recopie les information depuis le serveur DNS principal s'appelle un **serveur secondaire** ou **backup**. L'édition du fichier de la zone est faite sur le serveur principal qui envoie la version la plus récente du fichier au serveur DNS secondaire. Lorsqu'une machine envoie une requête au serveur secondaire, ce dernier y répond avec sa copie du fichier. Le fichier de zone du serveur secondaire a généralement une durée de vie (généralement de 24 heures). Si le serveur DNS primaire ne met pas à jour le fichier avant la période d'expiration, le serveur secondaire considère l'information comme dépassée. Si votre serveur DNS principal tombe en panne pendant quelques heures, vous n'aurez donc pas de problème. Les serveurs DNS secondaires peuvent être aussi nombreux que l'on le souhaite.

# Nom netbios

## Protocole NetBeui :

Windows 9x et NT pouvaient utiliser le protocole propriétaire Netbeui pour communiquer avec d'autre machine Windows.

Pour les réseaux de petite taille, une vingtaine de postes, cette solution permettait un partage simple des ressources. Les **applications NETBIOS** accédaient au réseau en s’appuyant sur le **protocole NETBEUI**.

Quelques définitions :

NetBIOS :

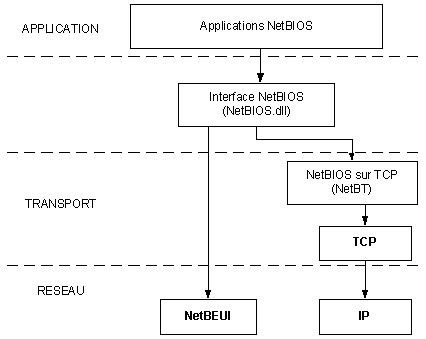
( Network Basic Input/output System) Interface de  programmation qui permet aux applications d’accéder au réseau local.   
NetBIOS utilise  un service  de noms pour contrôler les échanges de point à point.

**NetBEUI :**

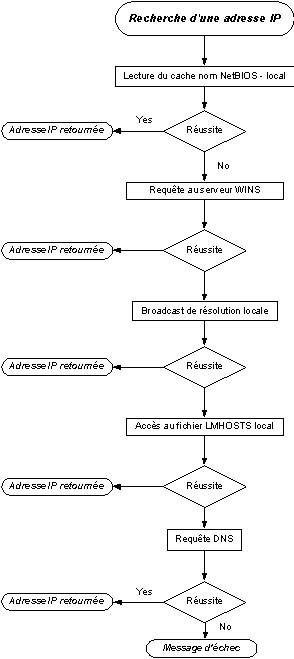
(NetBIOS Extended User Interface ) est le protocole de transport des réseaux Windows. Il ne peut pas être routé et repose  principalement sur les diffusions.

**NetBT**

(NetBIOS sur TCP/IP) est le service de résolution de noms NetBIOS pour les réseaux Windows sous TCP/IP.



## Résolution de nom NetBIOS

Windows peut utiliser différentes méthode pour effectuer la résolution de nom **netbios** :

1. NetBIOS name cache (vérifiable via nbtstat –n)
2. NetBIOS name server (WINS Il existe sous NT un serveur de nom NetBIOS connu sous l'appellation serveur WINS.)
3. IP subnet broadcasts (limité au sous-réseau)
4. Static Lmhosts file. (pour résoudre un nom netbios sur un autre réseau)
5. Static Hosts file (**optionnel** pour un nom d'hôte)
6. DNS servers (optionnel)

La manière dont Windows va résoudre les nom Netbios, dépends du paramétrage du poste, et de la configuration du réseau existant. Les différents modes de résolution suivants sont possibles, on parle de type de noeud:

1. **B-node (diffusion) :** utilise des broadcast pour l'enregistrement et la résolution des noms Netbios.
2. **P-node** : utilise un serveur de nom NetBios (Wins) pour l'enregistrement et la résolution des noms Netbios.
3. **M-node** : utilise des broadcast pour l'enregistrement. Pour la résolution, utilise d'abords des Broadcast, puis en l'absence de réponse passe ne mode P-node (donc utilise un serveur WINS)
4. **H-node** **(hybride)** : utilise un serveur de nom NetBios (Wins) pour l'enregistrement et la résolution des noms Netbios . Si un serveur ne peut pas être trouvé, il passe en b-node. (donc utilise des boradcast) . Il continue à chercher une serveur WINS et repasse en p-node des qu'il en trouve un disponible
5. **Microsoft-enhanced** : utilise les fichiers Lmhosts en plus des mode standard.

Par défaut, la plupart des clients sont paramétrés en B-nodes, c'est à dire émettent des broadcast...

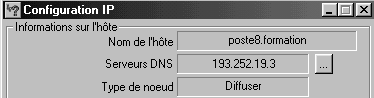
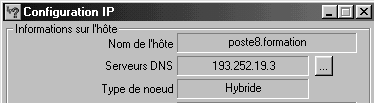
## Paramétrer la résolution NetBIOS

il est bien sûr possible de voir le mode de résolution actuellement en cours sur une machine avec **IPCONFIG /ALL** dans la rubrique "**type de noeud**"

* on peut facilement demander de passer de **B-nodes** à **h-nodes**, et vice-versa.

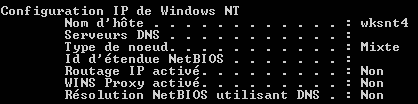
Il suffit de renseigner ou non l'adresse d'un serveur Wins sur le client...

serveur Wins **non** renseigné serveur Wins renseigné



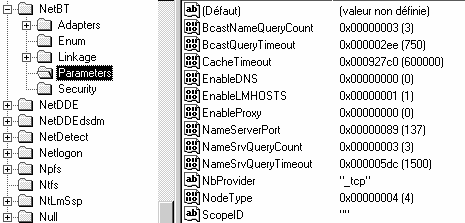
L'accès aux autres modes de résolution n'est possible que sur des machines NT ou 2000 (et ultérieurs):

* Par exemple, l'activation des **LmHosts** se fait dans les propriétés avancées de TCP-IP, onglets Wins.
* Par exemple le passage en Type de noeud M-Nodes,



ne peut se faire via modification de la base de registre par ajout d'une clé de type Dword dans l'entrée

**HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\NetBt\Parameters**



"type nœud"

Les valeurs possibles étant : **1 b-node diffuser**

**2 p-node homologues**

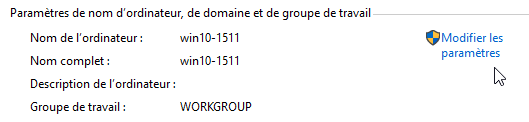
**4 m-node mélangé - mixte**

**8 h-node hybride**

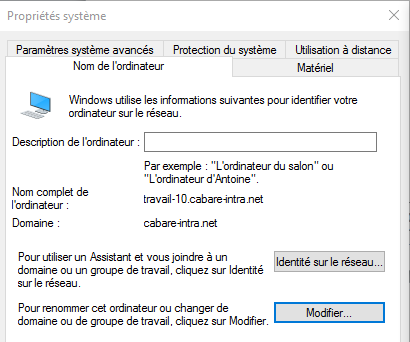
## Nom Netbios - Nom d'hôte:

A noter qu'un nom d'hôte et un nom d'ordinateur (nom netbios 15 caractères maxi, lettre chiffre tiret pas de différence à la casse) sont deux choses différentes, même si par défaut, dans un réseau microsoft, ce sont les mêmes. Depuis **Seven: Nom netbios** = **Nom d'hôte** Par défaut, il y a une traduction automatique !

Sous **windows 10** on demande **Modifier les paramètres**



Puis via **Modifier**

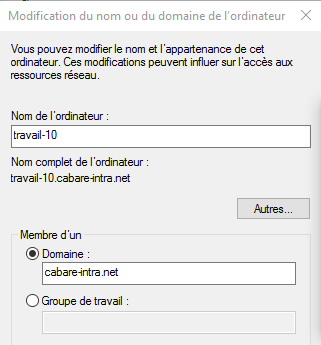
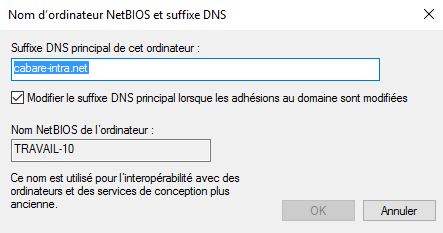


Apparition d'une zone   
"**Description de l'ordinateur**"

A ne pas confondre avec le nom de l'ordinateur

Accessible par **Modifier…**

Puis **Autres**



**N.B:** ne jamais rentrer une **Description de l'ordinateur** différente du **Nom de l'ordinateur (**ne jamais suivre l'exemple, mais utiliser les règles classiques (- de 15 caractères… etc…)

## Intérpretation des Nom NetBios :

**N.B:** On peut utiliser l'utilitaire **nbtstat** pour voir les noms NetBIOS avec la syntaxe suivante :

**nbtstat –n**

ou

**nbtstat –a nommachine**

**ou nbtstat –R** permet de purger les noms et force les réinscription depuis le fichier LMHOST

**ou nbtstat –c** permet de visualiser les noms présent dans le cache

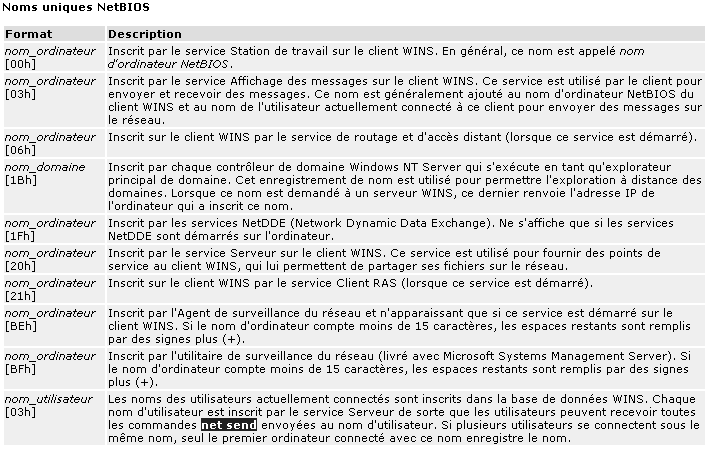
**ou nbtstat –RR** permet de purger les noms et force les réinscription sans avoir a redémarrer le poste (win XP)

Les **15 premiers caractères** d'un nom peuvent être spécifiés par un utilisateur. En revanche, le **16e caractère** du nom (hexadécimal 00-FF) indique toujours un type de ressource:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | **Nb (hexa)** | **Type** | **Usage** |
| <computername> | 00 | U | Workstation Service |
| <computername> | 01 | U | Messenger Service |
| -- MSBrowse -- | 01 | G | Domain Master Browser |
| <computername> | 03 | U | Messenger Service |
| <computername> | 06 | U | RAS Server Service |
| <computername> | 1F | U | NetDDE Service |
| <computername> | 20 | U | File Server Service |
| <computername> | 21 | U | RAS Client Service |
| <computername> | 22 | U | Microsoft Exchange Connector |
| <computername | 23 | U | Microsoft Exchange Store |
| <computername> | 24 | U | Microsoft Exchange Directory |
| <computername> | 30 | U | Modem Sharing Server Service |
| <computername> | 31 | U | Modem Sharing Client Service |
| <computername> | 43 | U | SMS Clients Remote Control |
| <computername> | 44 | U | SMS Admin Remote Control Tool |
| <computername> | 45 | U | SMS Clients Remote Chat |
| <computername> | 46 | U | SMS Clients Remote Transfer |
| <computername> | 4C | U | DEC TCPIP service on NT |
| <computername> | 42 | U | mccaffee anti-virus |
| <computername> | 52 | U | DEC TCPIP service on NT |
| <computername> | 87 | U | Microsoft Exchange MTA |
| <computername> | 6A | U | Microsoft Exchange IMC |
| <computername> | BE | U | Network Monitor Agent |
| <computername> | BF | U | Network Monitor Application |
| <username> | 03 | U | Messenger Service |
| <domain> | 0 | G | Domain Name |
| <domain> | 1B | U | Domain Master Browser |
| <domain> | 1C | G | Domain Controllers |
| <domain> | 1D | U | Master Browser |
| <domain> | 1E | G | Browser Service Elections |
| <INet~Services> | 1C | G | IIS |
| <IS~computer name> | 00 | U | IIS |
| <computername> | [2B] | U | Lotus Notes Server Service |

**Il existe essentiellement 2 groupes**

**Unique (U):** Utilisé pour associer l'ordinateur par son nom à une adresse IP unique. Avec ce type de nom, trois types d'enregistrements sont ajoutés statiquement à la base de données WINS pour le nom d'ordinateur spécifié. Les types [00h]**WorkStation**, [03h]**Messenger** et [20h] **Serveur de fichiers**.



**Group (G):** Appelé aussi groupe ordinaire. Avec ce type, l'adresse IP de l'ordinateur n'est pas stockée dans WINS, mais résolue par le biais des diffusions du sous-réseau local.



enfin, moins important

**Multihomed (M):** Utilisé pour inscrire un nom unique pour un ordinateur ayant plusieurs adresses IP (plusieurs cartes utilisant chacune une adresse unique ou une seule carte réseau configurée avec plusieurs adresses IP).

**Domain Name (D):** Indique une entrée mappée de *nom de domaine* [1C] pour la localisation des contrôleurs de domaine Windows NT

# Hosts - LMhosts

## Fichier Hosts et LMHosts:

Un fichier **HOSTS** permet d'établir un mappage entre une adresse IP et un nom de machine (nom d'hôte), c'est un fichier issu du monde unix. L'alternative au fichier hosts est un serveur DNS.

A utiliser lorsque : on souhaite effectuer des transactions IP (ping, ftp…) lorsque la machine à atteindre n'a pas eut son nom résolu par DNS. Permet donc d'être sûr de trouver un poste, indépendamment du fonctionnel d'un serveur DNS.

Un fichier **LMHOSTS** permet également d'établir un mappage entre une adresse IP et un nom de machine (nom d'ordinateur ou nom netbios). L'alternative au fichier LMHOSTS est le service WINS. Le fichier LMHOSTS (Lan Manager HOSTS) concerne essentiellement les réseaux Microsoft.

A utiliser lorsque : on souhaite effectuer des transactions réseau microsoft, (Lan Manager commande net…, mécanisme de voisinage réseau…) lorsque la machine à atteindre ne fait pas partie du même sous-réseau, et qu'il n'y a pas de serveurs WINS opérationnel. Permet donc d'être sûr de trouver un poste, indépendamment du fonctionnel d'un serveur WINS.

## Fichier lmHosts (nom netbios):

Situé pour les postes NT – 2000 en **WINNT\SYSTEM32\DRIVERS\ETC**

Un exemple est fournit avec le fichier **lmhosts.sam** avec une extension .sam pour sample qu'il faut évidemment enlever pour rendre actif le fichier **lmhosts**. Il permet de solutionner un nom netbios sur un autre sous-réseau.

# Ce fichier est compatible avec les fichiers lmhosts de Microsoft LAN

# Manager 2.x TCP/IP et les extensions offertes sont les suivantes:

# #PRE

# #DOM:<domaine>

# #INCLUDE <nom\_de\_fichier>

# #BEGIN\_ALTERNATE

# #END\_ALTERNATE

# \0xnn (caractère non imprimable)

Donc un fichier lmhost peut contenir une ligne du genre

**192.168.1.1 NOMPOSTE #PRE**

avec 192.168.1.1 l'adresse ip du POSTE

avec NOMPOSTE le nom NETBIOS du POSTE

Après modification du fichier **lmhosts** il faut impérativement redémarrer le poste, ou faire une commande en ligne

**Nbtstat –R** (avec le R majuscule...)

Pui vérifier la prise ne compte avec un

**Nbtstat –c** (avec le c minuscule...)

### Détails écriture lmhosts

1. **10.0.0.1 PDCName #PRE #DOM:Domain-name**
2. **10.0.0.1 "Domain-name \0x1b" #PRE**

**N.B:** L'espacement de ces entrées est obligatoire. Remplacez 10.0.0.1 par l'adresse IP de votre contrôleur principal de domaine, PDCName par le nom NetBIOS de votre contrôleur principal de domaine, et Domaine par le nom de domaine de Windows. Au total il doit y avoir 20 caractères à l'intérieur des guillemets (le nom de domaine, + le nombre d'espaces appropriés pour obtenir 15 caractères, + la barre oblique inverse, + la représentation hexadécimale NetBIOS du type de service).

**N.B:** Pour déterminer l'emplacement du 16e caractère, copiez la ligne suivante dans votre fichier LMHOSTS :

**# Adresse IP "123456789012345\*7890"**

Alignez les guillemets doubles (") en ajoutant ou supprimant des espaces dans la ligne de commentaire, et placez la barre oblique inverse sur la 16e colonne (marquée d'une astérisque). N'utilisez de tabulation mais des ESPACES après le nom et avant la barre oblique inverse (\).

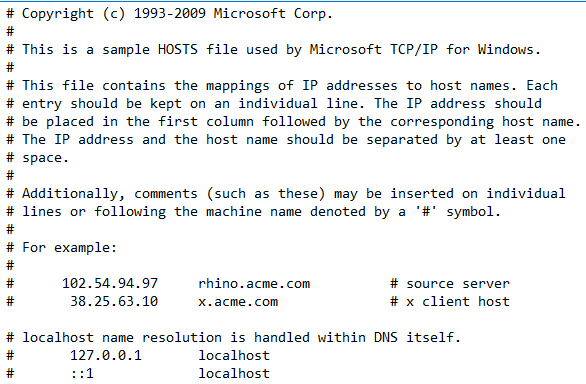
**NB**: Attention, le fichier contient toujours une ligne blanche vide à la fin !

## Fichier hosts (nom d'hôte):

Un exemple est fourni sur les machines avec le fichier **hosts**

Il permet de solutionner un nom d'hôte.





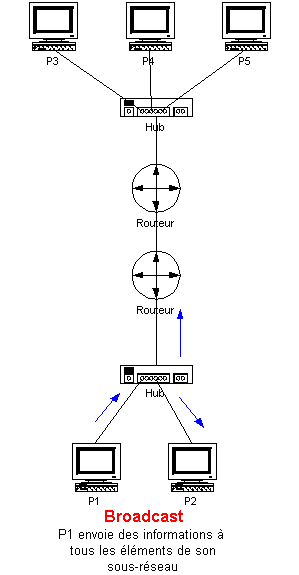
**NB**: Attention, le fichier contient toujours une ligne blanche vide à la fin !

# annexe : Trames tcp/IP

## Broadcast :

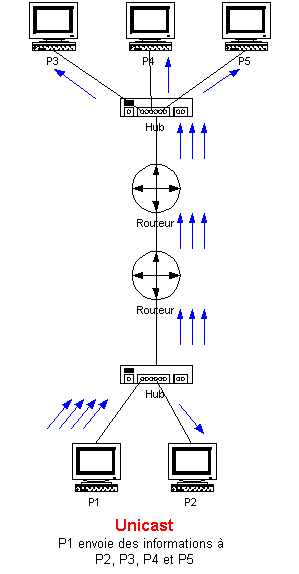
Le principe du **broadcast** est d'envoyer une information à tous les ordinateurs du réseau où l'on est. Au lieu d’envoyer en unicast vers l’adresse IP de la chaque machine (ex. 193.169.1.37 avec un masque 255.255.255.0),

L'adresse de **broadcast** est une adresse IP qui termine en .255 dans des réseaux de classe A, B ou C, cette adresse est celle qui permet de faire de la diffusion à toutes les machines du réseau

On envoie la trame à tous les ordinateurs du sous-réseau en utilisant l'adresse de **broadcast** (ici, 193.169.1.255). Cette adresse est réservée à cet usage. Chacun des ordinateurs du sous-réseau regarde et traite la trame comme si elle leur était personnellement adressée.

Les trames de **broadcast** ont une caractéristique particulière : c’est de ne pas pouvoir passer les routeurs puisqu'il s'adresse uniquement à tous les ordinateurs d'un même sous-réseau.

## Unicast :

C’est le principe le plus utilisé et le plus simple. Les ordinateurs possédant chacun une adresse IP, on peut envoyer les trames en spécifiant l’adresse IP de l’ordinateur à qui on veut envoyer les informations. Les éléments actifs et passifs du réseau ( commutateurs, répéteurs, routeurs, ... ) dirigent l'information dans la bonne direction pour que les trames arrivent au bon endroit. Seule la machine ayant l’adresse contenue dans la trame regarde et traite l’information.

Il existe 3 classes d'adresses unicast :

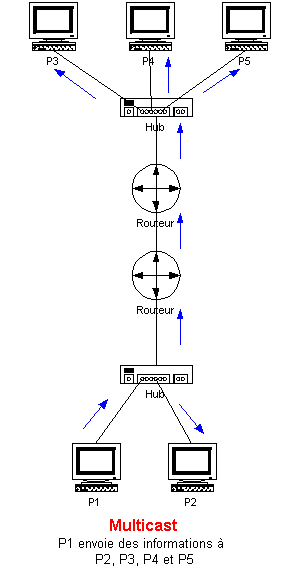
La classe A : Adresses comprises entre 1.0.0.x et 127.255.255.x

La classe B : Adresses comprises entre 128.0.0.x et 191.255.255.x

La classe C : Adresses comprises entre 192.0.0.x et 223.255.255.x

## Multicast :

Plutôt que d’envoyer les fichiers du serveur vers chacune des machines clientes (unicast) on peut n’envoyer l’information qu’une seule fois et chaque ordinateur client la récupère. En effet, dans un réseau Ethernet par exemple, toutes les trames qui circulent passent par tous les ordinateurs. C’est le principe du multicast : on envoie l’information à une adresse et tous les clients écoutent cette adresse.



Chaque client multicast s’enregistre avec une adresse IP multicast de classe D (entre 224.0.0.0 et 239.255.255.255 sauf 224.0.0.0 non utilisée et 224.0.0.1 qui correspond au "broadcast du multicast"). C’est sur cette adresse que les informations vont être envoyées.

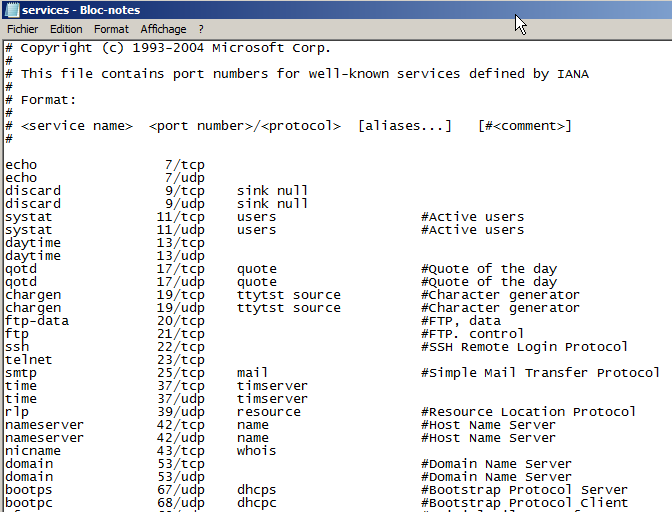
Les clients écoutent ce qui arrive sur cette adresse et suivent la procédure décrite par le protocole multicast implémenté.

# Dossier ..\System32\driver\etc

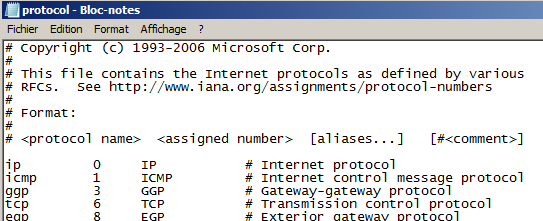
## Fichiers exemples Windows :

Depuis Windows Seven, on peut trouver dans les machines Windows un "mémo" stocké dans le dossier d’installation de l’OS **%system%\system32\driver\etc**

Sur les N° de port : fichier **service**



sur les N° de protocole fichier **protocol**



Ainsi que des exemples de fichier host et lmhost…

# TP - WorkGroup entre réseaux

## 1 réseau IP et x Workgroups différents:

on donne à des machines faisant partie de différents réseau des adresses en classe C privée dans un seul réseau :

id réseau **192.168.1** donc

adresse **192.168.1**.1 pour la 1° ,

adresse **192.168.1**.2 pour la 2° ,

adresse **192.168.1**.X pour la X° ,

masque 255.255.255.0

Pour certaines machines, on donne un workgroup d’appartenance « grpdroit »,

à d’autres on donne un workgroup d’appartenance « grpgauche »,

## test et vérification :

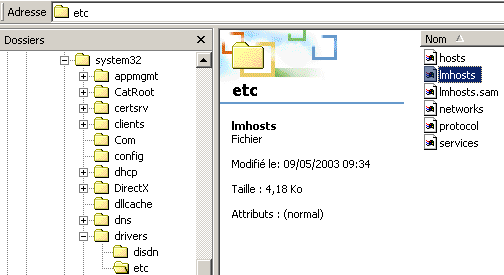
Vérifier la communication entre les machines ?, les workgroups ?

Que se passe-t-il et pourquoi ?

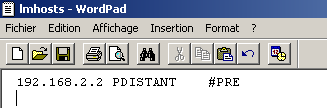
# TP - Modifier lmhosts

## Inscrire une machine simple dans lmhosts :

Inscrivons le poste nommé **pdistant** d'adresse **192.168.2.2** dans la table préchargée de résolution de nom netbios d'une machine:



Il suffit d'éditer le fichier texte et d'y inscrire la ligne suivante



On recharge par la commande **nbtstat –R**



et on visualise par la commande **nbtstat –c**:



## Inscrire un Contrôleur de Domaine dans lmhosts :

En général, on n'a pas besoin d'inscrire des postes génériques, mais plutôt un contrôleur de domaine…Dans ce cas la ligne se complique un petit peu puisqu'il est nécessaire d'indiquer le nom de domaine en plus…. Il faut effectuer donc 2 entrées, une pour le PDC et l'autre pour le nom de domaine.

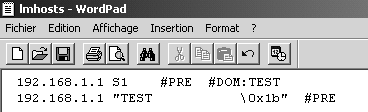
**10.0.0.1 PDCName #PRE #DOM:Domain-name**

**10.0.0.1 "Domain-name \0x1b" #PRE**

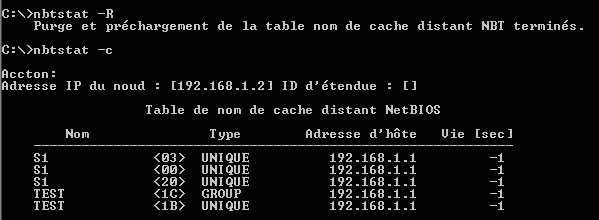
**N.B** : Le nom de domaine dans cette entrée respecte la casse.

**N.B:** L'espacement de ces entrées est obligatoire. Remplacez **10.0.0.1** par l'adresse IP de votre contrôleur principal de domaine, **PDCName** par le nom NetBIOS de votre contrôleur principal de domaine, **Domain** par le nom de domaine de Windows

Inscrivons le contrôleur de domaine **TEST** nommé **S1** d'adresse **192.168.1.1** dans la table préchargée de résolution de nom netbios d'une machine:



avec pour vérification



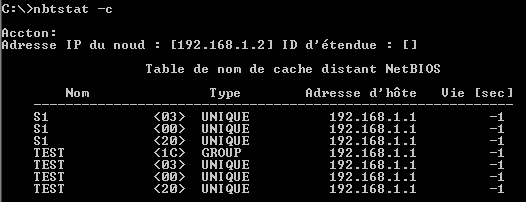
**NB**: Au total il doit y avoir 20 caractères à l'intérieur des guillemets   
(le nom de domaine, +   
le nombre d'espaces appropriés pour obtenir 15 caractères, +   
la barre oblique inverse, +  
la représentation hexadécimale NetBIOS du type de service).

**NB**: Attention, le fichier contient toujours une ligne blanche vide à la fin !

Ainsi une simple erreur de nombre de caractère (différent de 20 ici)



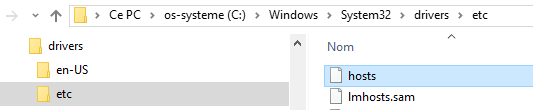
ne génère aucun message d'erreur, mais simplement une mauvaise inscription :



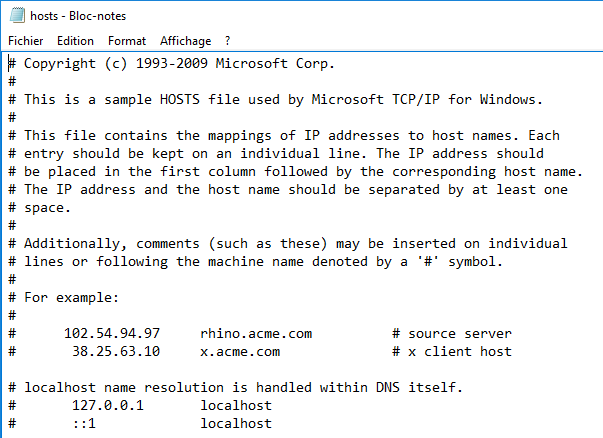
# TP - Modifier hosts

## Inscrire une machine dans hosts :

Le fichier est fourni directement dans les postes Windows, en **%system%\system32\driver\etc**



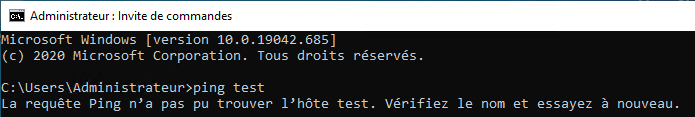
On peut noter que les boucles locales IPV4 et IPV6 ne sont plus gérées dans le Hosts



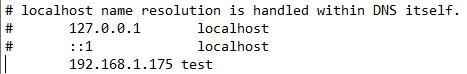
**N.B** : pour éditer le fichier **host**, il faut penser à plusieurs choses

* Vérivier s'il est noté avec l’attribut lecture seule
* Il faut vérifier que l’on a bien les droits en accès
* Il faut d’abords lancer le bloc note en tant qu’administrateur et ouvrir le fichier. (si on demande depuis le fichier de l’ouvrir avec le bloc note, si l’UAC est configuré, on ne pourra pas l’enregistrer..
* Redémarrer le poste

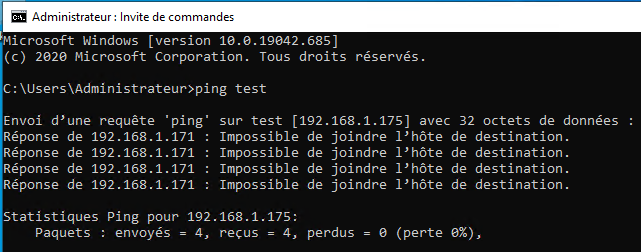
Faisons un **ping** sur une machine inexistante ***test***, on obtient



Inscrivons le poste nommé **test** d'adresse **192.168.1.175** (une adresse fictive) dans la table de résolution locale, puis reboot du poste



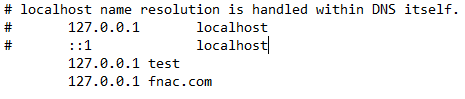
Du coup si avant on n'avait aucune possibilité de faire un **ping test** désormais, on peut désormais au moins envoyer la trame… (Évidemment le retour est plus… délicat !)



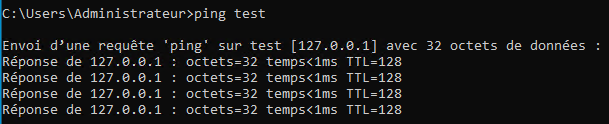
**N.B**: Si une réponse est émise, cela ne veut pas dire que cette machine ***test*** existe, cela veut dire qu’une machine 192.168.1.175 a répondu !

## Interdire une machine un site dans hosts :

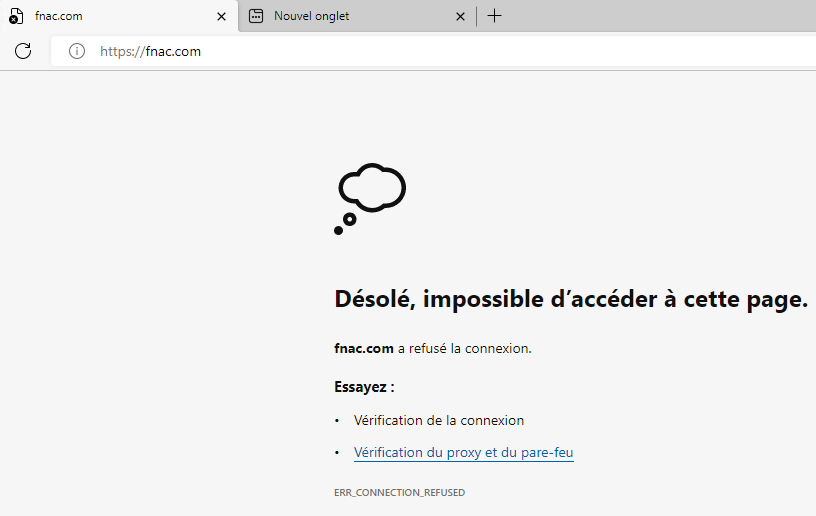
**N.B**: Un moyen simple d'invalider un nom consiste à le renvoyer sur l'adresse de bouclage **127.0.0.1, idem pour une URL d'un site à proscrire**



donnera



Et une tentative d'accès sur le site de la ***Fnac*** donnera

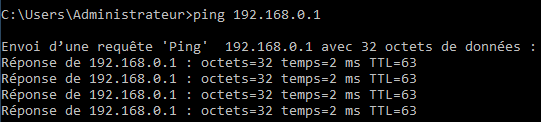


# Tester TCP/IP

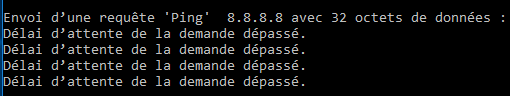
## ICMP et l'Utilitaire PING :

Permet d'envoyer une trame IP de test vers une machine,

### Types de réponses à un ping

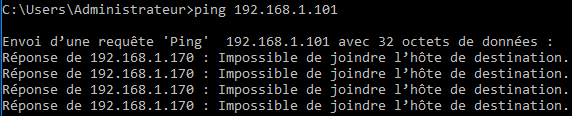


"***Réponse de … octets=32 temps= TTl***" indique que 4 trames de 32 octets ont été aquittées par l'adresse IP de destination, avec un temps et le TTL:



"***Délai d'attente de la demande dépassé***" signifie généralement qu'aucune réponse ICMP n'a été reçue par la machine qui lance le ping. Plusieurs causes possibles :

* la cible est configurée pour ne pas envoyer d'ICMP Reply,
* il y a au moins un firewall qui bloque ces ICMP entre la cible et la station qui lance le ping,
* effectivement la cible répond avec un ICMP qui possède un TTL trop petit et donc le paquet est droppé quelque part.



"***Impossible de joindre l'hôte de destination***" devrait normalement être le cas où le réseau IP destination existe mais l'adresse IP cible ne répond pas. Plusieurs causes possibles:

* La pile IP de la cible n'est pas active
* la cible est configurée pour ne pas envoyer d'ICMP, etc.

La différence entre "délai d'attente dépassé" et "impossible de joindre l'hôte", c'est que dans le 1er cas, la station qui lance le ping ne reçoit rien, tandis que dans le 2ème cas de figure, il reçoit quelque chose...

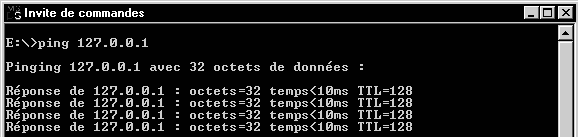


"***échec de la transmission. Défaillance générale***" devrait normalement être le cas où la trâme IP ne peut partir. Plusieurs causes possibles:

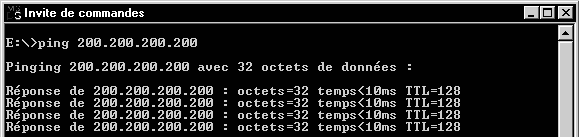
* Paramétrage IP erroné
* Filtrage IP par un élément externe sur le réseau

### Méthodologie de test

En tapant **Ping 127.0.0.1** si on ne reçoit pas les 4 lignes suivantes, cela veut dire que la pile TCP/IP n’est pas installée correctement

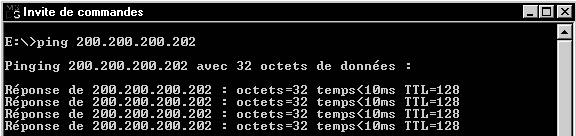


En tapant **Ping XX.XX.XX.XX** avec l’adresse de notre propre station depuis laquelle on « pingue », si on ne reçoit pas les 4 lignes suivantes, cela veut dire que l’adresse de la station est erronée



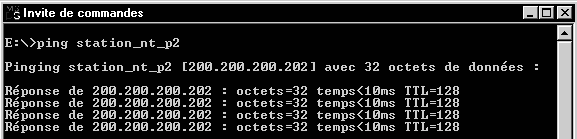
Jusqu'à présent on n’a rien envoyé sur le réseau, maintenant considérer que notre poste est correctement configuré sous TCP/IP, on va utiliser le réseau

En tapant **Ping XX.XX.XX.XX** avec l’adresse de la station que l’on souhaite atteindre, si on ne reçoit pas les 4 lignes suivantes, cela veut dire soit que l’adresse de la station est erronée soit que la connectique est mauvaise



En tapant **Ping NOMSTATION (peu conseillé)** avec le nom d'hôte à atteindre, si on ne reçoit pas les 4 lignes suivantes, cela veut dire que le nom est erroné, ou qu'il n'est pas dans le même réseau **IP** s il n'y a pas de **DNS** . (Les **broadcasts** ne sont pas routable)

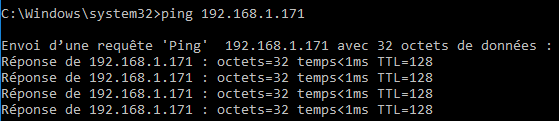
S'il n'y a pas de **DNS**, la résolution de nom se fera par des mécanismes de **broadcass** **netbios**



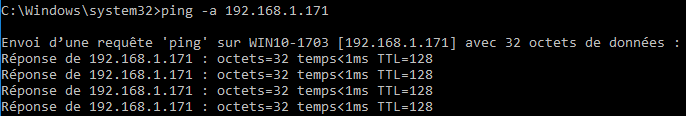
### Ping –a

On peut aussi taper **Ping –a XX.XX.XX.XX.**

Le nom de la station que l’on souhaite atteindre sera résolu en même temps que le retour de trame, ce qui permet de connaître en cas de problème le nom renvoyé par la machine...



L’option **–a** force la résolution de nom ici ***WIN10-1703***



### Ping –t

Une option intéressante est présente est **–t** dans **Ping XX.XX.XX.XX –t**

****

avec la combinaison de touche **CTRL + Attn** pour afficher les statistiques

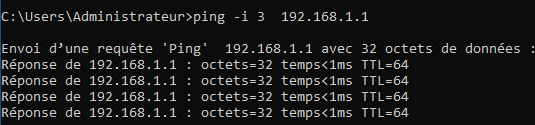


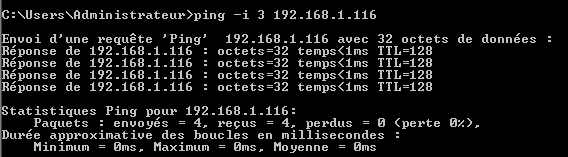
avec la combinaison de touche **CTRL + C** pour arrêter la commande

## Test TTL ping -i:

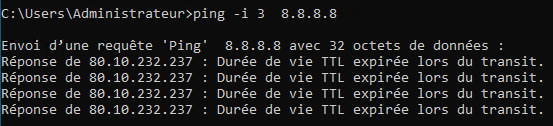
L'option **–i x p**ermet de spécifier la valeur **Time to Live**

Si un TTL très court à 3 permet d’atteindre la machine d’a coté

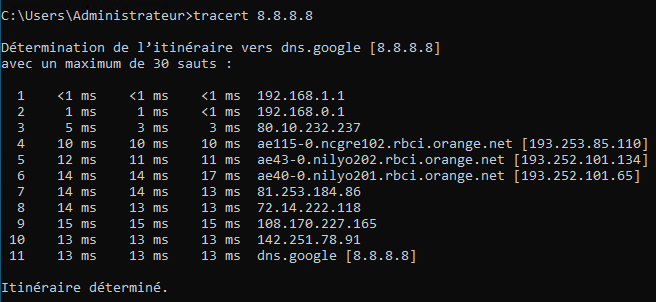




Le même TTL à 3 permet ne permet pas d’atteindre une machine distante, telle que un serveur DNS de Google en 8.8.8.8



Ce message en provenance du 3° routeur indique la mise à mort de la requête ! Car en effet ici il lui faudrait 11 sauts…



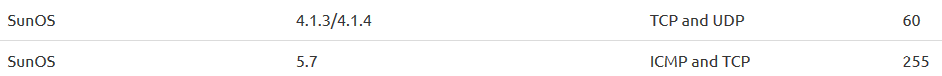
Le TTL d'une machine Windows est par défaut à 128

Le TTL d'une machine Linux est par défaut à 64

D'autres OS ont d'autres valeurs par défaut

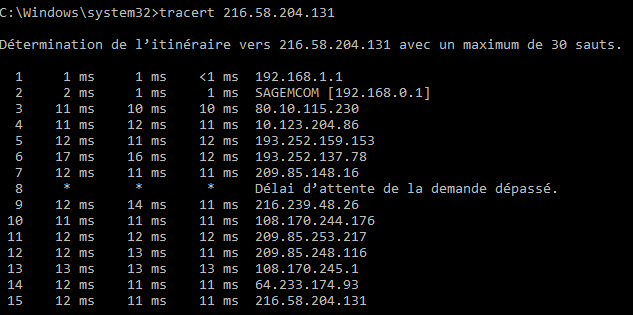




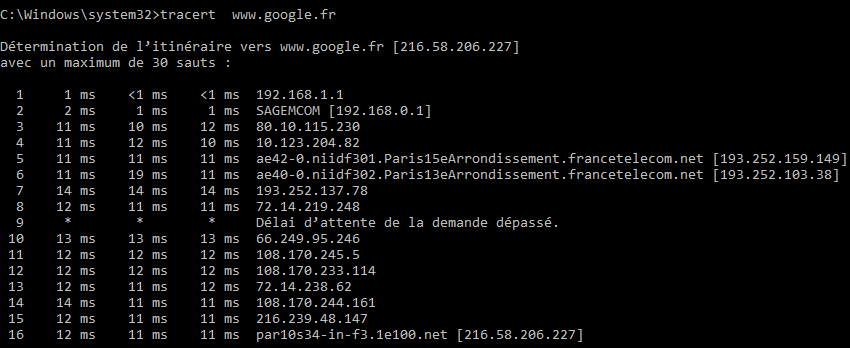


## Tracert :

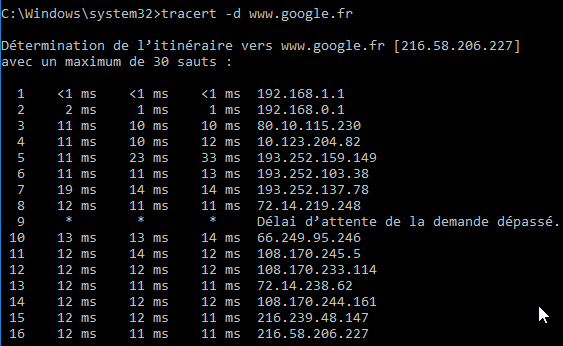
En tapant **tracert xx.xx.xx.xx** on demande de tracer la route pour atteindre une adresse ip



On peut aussi si une résolution DNS est présente, (possible) utiliser un nom



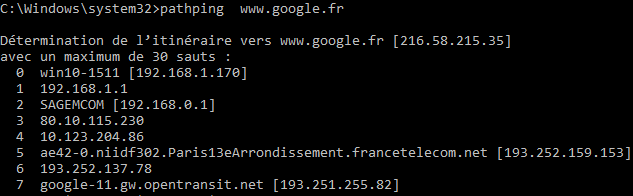
**N.B** : comme de nos jours les routeurs ne donnent souvent plus leur nom, on peut accélérer le temps de réponse (ne pas attendre une résolution de nom qui, ne viendra pas dans 99% des cas), en tapant l’option **–d**

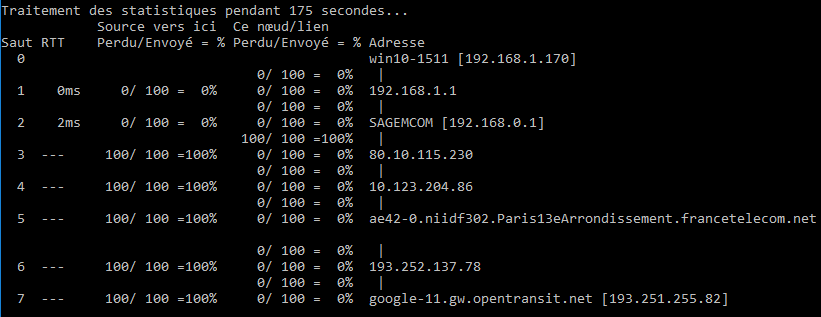
\*

## Pathping :

Depuis windows 2000 une commande combinée existe **PATHPING**

Elle commence à faire le même travail qu’un **tracert**, mais ensuite elle donnera un certain nombre de statistiques (un peu comme **ping /t**)

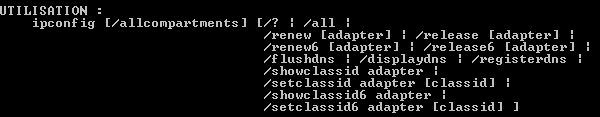
avec dans un 2° temps ( 25secondes par nœud)



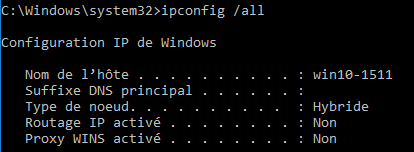
* Le nom de la première machine de départ est donné
* Une perte de paquet régulière supérieure ou égale à 1 % sur un routeur indique un défaut

## Ipconfig.exe /all:

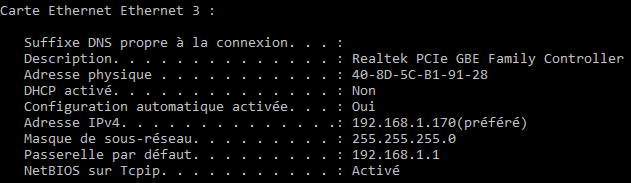
Sous Windows **Ipconfig.exe** depuis une boite dos ou une invite système



L'affichage complet **ipconfig /all** donne le nom d'hôte, le type de nœud de résolution des noms netbios, et si le routage est activé sur le poste



ensuite pour chaque carte réseau, on récupère son adresse mac, si elle est en client dhcp ou en IP statique, son adressage IP complet, et si Netbios Over TCp-IP est maintenu ou non.



Il peut y a voir beacoup de cartes réseaux….

## ARP et l'Utilitaire ARP –a :

Les essais sur une configuration peuvent se faire à bas niveau

**Permet de connaître l'adresse physique d'une machine**



**ARP** est un protocole permettant la résolution adresse Ip => adresse physique. ARP est mis en œuvre automatiquement lors de toute requête IP, et typiquement lors d'un **ping**....

### Arp –a

En tapant **ARP -a** on affiche le contenu du cache actuellement présent sur notre machine. Sur une machine que l'on démarre, le cache peut être vide.



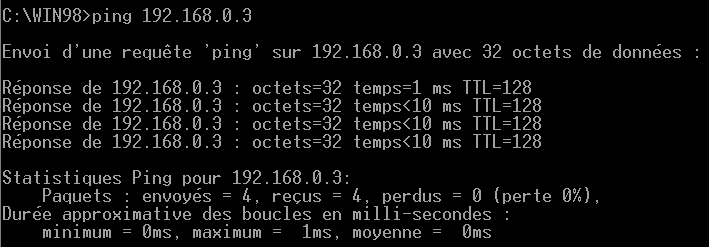
après un coup de voisinage réseau, le master browse ayant répondu, le cache contient désormais son adresse IP et son adresse physique



si on attend, le cache va finir par se vider et de nouveau on aura



Si on fait un **ping** sur une machine donnée, alors son "entrée" dans la table est effectuée des que la réponse est obtenue...



ce qui donne ensuite



un **F5** (pour rafraîchir l'écran du voisinage réseau) provoquerait alors une autre entrée dans le cache ARP...etc, etc...



#### Ajout –suppresssion d’entrée Arp

On peut rentrer une adresse statique. **ARP –s**



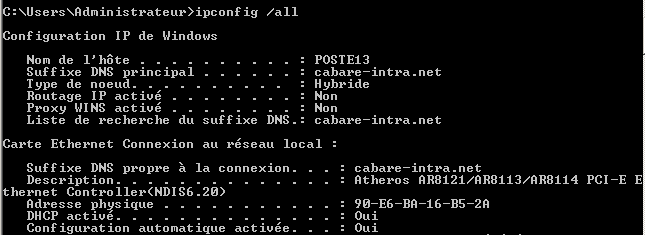
ce qui donnerait dans la table l'aspect suivant



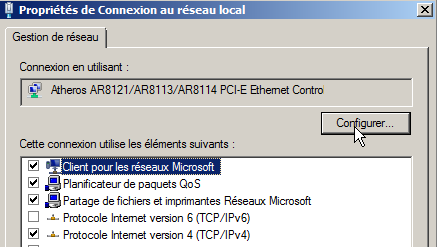
Cette entrée "statique" ne sera purgée de la table que lors d'un redémarrage du poste. Si on souhaite la modifier il suffit de rentrer de nouveau une commande du type **arp –s**

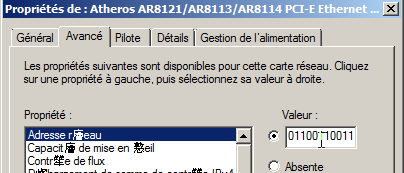
## Usurpation d’adresse ARP :

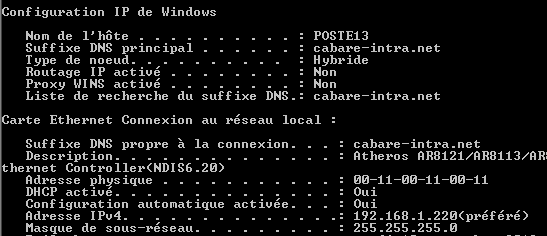
Soit une adresse donnée pour une carte



Si le driver le permet, il est facile sous windows d’usurper l’adresse mac en demandant sur la carte réseau **Configurer…**

 et on choisit **Adresse Réseau**



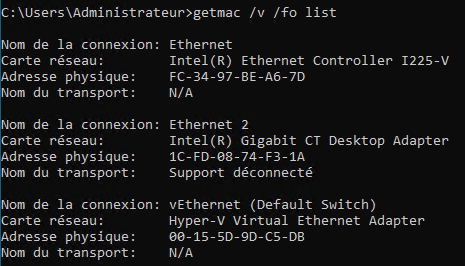


## Getmac /v:

On peut aussi taper la commande **getmac.** L’option **/V** est la plus utilisée



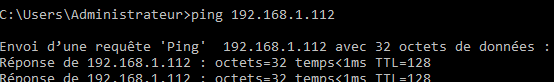
Si on veut un affichage plus complet, on ajoute **/fo list**,

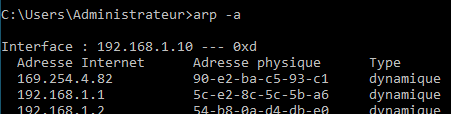


## A distance = ping + arp / nbtstat -A:

Pour obtenir une **adresse mac** d’une machine distante, par exemple en 192.168.1.112 – poste-12, le plus simple c’est d’utiliser une des 2 méthodes suivantes :

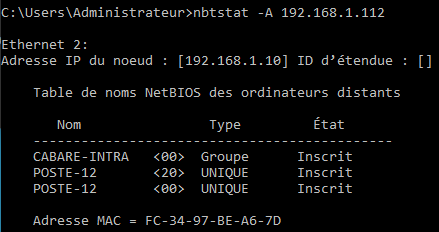
* Un **ping** suivit de la commande **arp –a** comme dans :





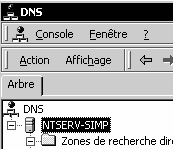


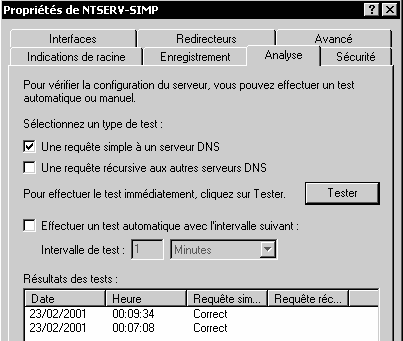
* Une commande **nbtstat** avec l’option **–A** et l’adresse de la machine distante



# Test de DNS

## Test DNS d’un client d’un domaine :

La bonne marche du serveur DNS peut se tester via les propriétés du Serveur DNS dans la console MMC de gestion du DNS



Onglet **Analyse**

La bonne marche des enregistrements dans le DNS peut se tester via la commande **Nslookup**

Cet outil de diagnostic affiche des informations sur les serveurs de noms DNS (système de noms de domaine). **Nslookup** est disponible uniquement si le protocole TCP/IP est installé.

### Nom d’hôte et FQDN

Soit un domaine ***cabare-intra.net***, et un poste nommé ***poste-10*** appartenant à ce domaine. On appellera

* nom d’hôte ***poste-10***
* FQDN Fully qualified Domain name ***poste-10.cabare-intra.net***

De manière générale, lorsque l’on fait les tests d’un serveur DNS d’un domaine, depuis une machine du domaine, il est suffisant d’utiliser le nom d’hôte, mais si on effectue un test de DNS depuis une machine ne faisant pas partie du domaine il est alors nécessaire d’utiliser le FQDN0

### Nslookup en mode interactif

**Nslookup** propose deux modes : interactif et non interactif.

On passe en mode inter-actif en tapant simplement **nslookup**,

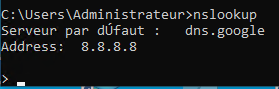
On sortira du mode inter-actif en tapant **exit**.

**mode interactif 1° (hors domaine – avec google – 8.8.8.8)**

* En premier argument, tapez le nom ou l'adresse IP de l'ordinateur pour lequel la recherche est effectuée.
* En deuxième argument, tapez le nom ou l'adresse IP d'un serveur de noms DNS. (Si omis, le serveur de noms DNS par défaut est utilisé)

Dans les exemples ci-dessous, on est relié au **DNS** de Google : **8.8.8.8**

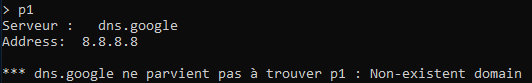
un **nslookup** donnera



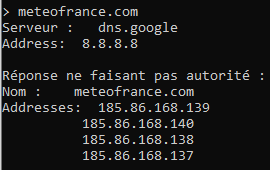
Au prompt de la commande **nslookup** **">"**,il faut taper des résolutions à satisfaire… , jusqu'à ce que l'on en sorte, par **exit**



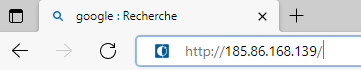
Recherche d'un nom inconnu, par exemple ***p1***

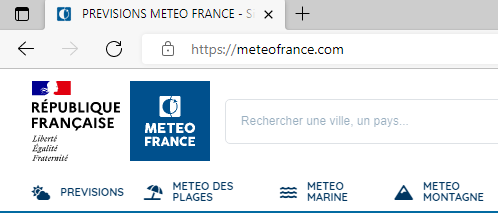


Recherche d'un nom connu, par exemple ***meteofrance.com***



Ce qui devrait permettre une opération du genre

 qui donnera



**mode interactif 1° (avec domaine)**

* En premier argument, tapez le nom ou l'adresse IP de l'ordinateur pour lequel la recherche est effectuée.
* En deuxième argument, tapez le nom ou l'adresse IP d'un serveur de noms DNS. (Si omis, le serveur de noms DNS par défaut est utilisé)

Dans les exemples ci-dessous,

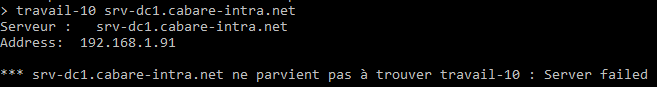
un client correct se nomme "**travail-10"**,

le serveur DNS par défaut est le serveur "**srv-dc1"**

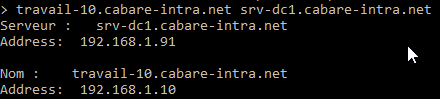
Un client incorrect se nomme "**poste-x"**



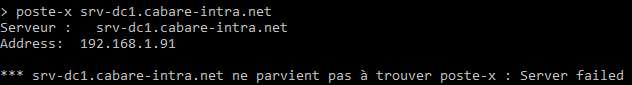
Ici on ne trouve pas le serveur DNS , il faut un FQDN ?



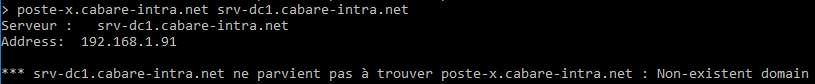
Ici on ne trouve pas de résolution pour le nom de ce client il faut un FQDN ?



Ici tout est parfaitement résolu



Ici on ne trouve pas de résolution pour le nom de ce client il faut un FQDN ?



Ici on ne trouve pas ce client dans le domaine, il n’y a pas d’erreur !!!

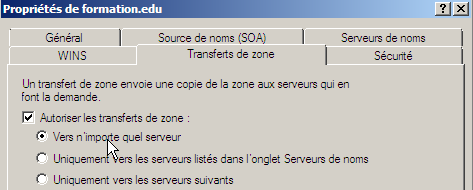
**mode interactif 2°**

**nslookup** accepte une autre commande en mode interactif, permettant de liste tous les enregistrement SRV présents dans le DNS.

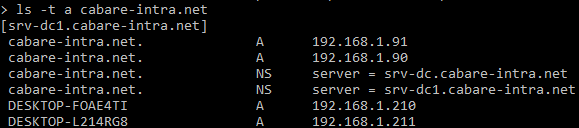
**N.B**: un certain type de requête peut être inhibée par défaut Pour que ces commandes soient possibles, il faut en effet que le transfert de zone sur le serveur DNS soit autorisé. Par défaut depuis 2008 les transferts ne sont pas autorisés (pour des raisons de sécurité)



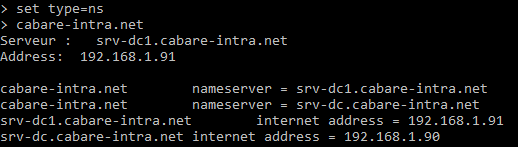
Pour la zone en question, on demande **propriétés**, puis **Autoriser les transferts**



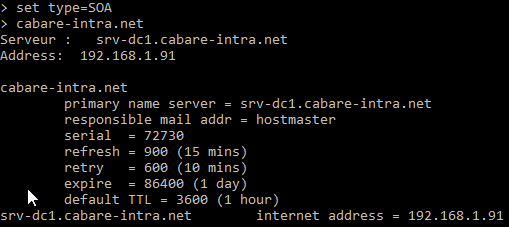
avec **ls -t a** suivi de **nomdomaine** on obtient tous les enregistrement **A Hôtes du Domaine**



avec **set type=NS** suivit de **nomdomaine** on obtient tous les SRV correspondant a des **NS name server**



avec **set type=SOA** suivit de nomdomaine on obtient tous les SRV correspondant a des **SOA Start of Authority**



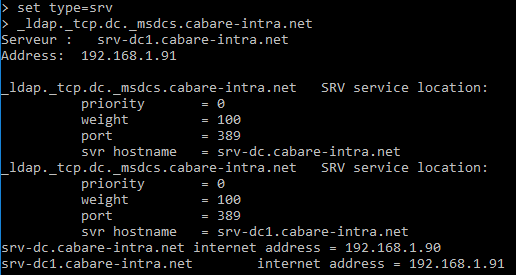
**mode interactif 3°**

Pour vérifier l'enregistrement DNS pour tous les contrôleurs de domaine à l'invite nslookup (">"), tapez :

**set type=SRV** suivit de **\_ldap.\_tcp.dc.\_msdcs.****nomdomaine**

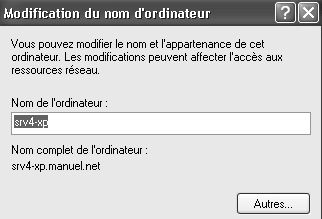
où **nomdomaine** est le nom DNS configuré pour être utilisé avec votre domaine Active Directory et tout contrôleur de domaine qui lui est associé.

Dans l'exemple, si le nom de domaine DNS de votre domaine est **domaine1.edu**, tapez **\_ldap.\_tcp.dc.\_msdcs.domaine1.edu**

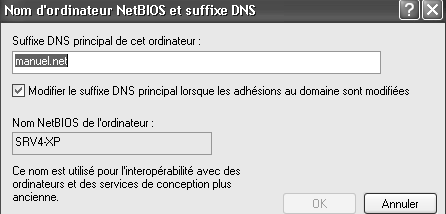


## Nslookup et non-réponse de Serveur Windows :

Il peut exister un problème d’interrogation de serveur 2008 depuis un poste windows ne faisant pas partie d’un domaine. Cela arrive lorsqu’il y a des différences entre les noms netbios - nom d’hôtes des postes- ainsi que le nom DNS du domaine dont on interroge le serveur. Pour solutionner cela on peut vérifier que notre machine, dans l’onglet identification du poste de travail

 **Autres…**

Indique bien que le DNS de rattachement est celui que l’on souhaite interroger…



## Nslookup et Ping :

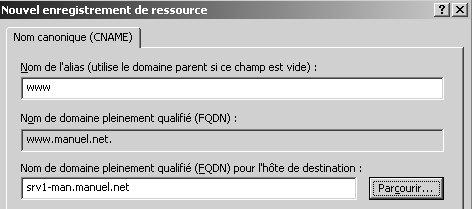
Il ne faut pas confondre les deux outils. Créons un hôte **bidon** dans le DNS



On pourra le tester avec **nslookup**,

Mais pas avec **ping**….

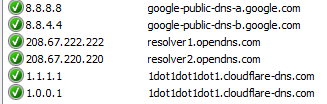
Créons un **Alias** « www » sur une machine existante (notre serveur)



On pourra le tester avec **nslookup**, et ici aussi utiliser **ping !**

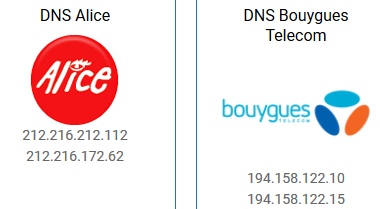
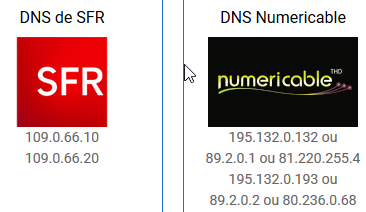
## Serveur DNS public – connus :

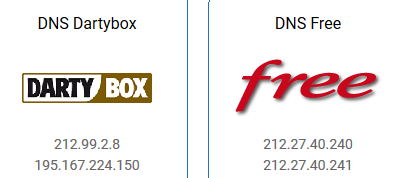
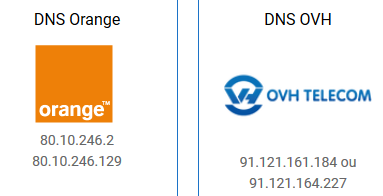
Si un certain nombre de Serveurs DNS public existent



On pourrait rajouter DNS Watch 84.200.69.80 et 84.200.70.40 ...

La liste des DNS des FAI est évidemment… abondante, quelques exemples

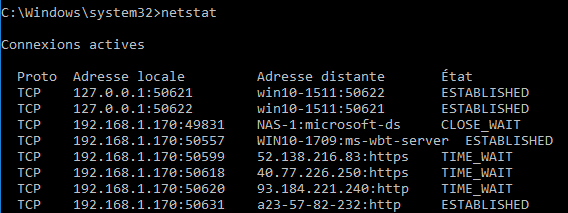




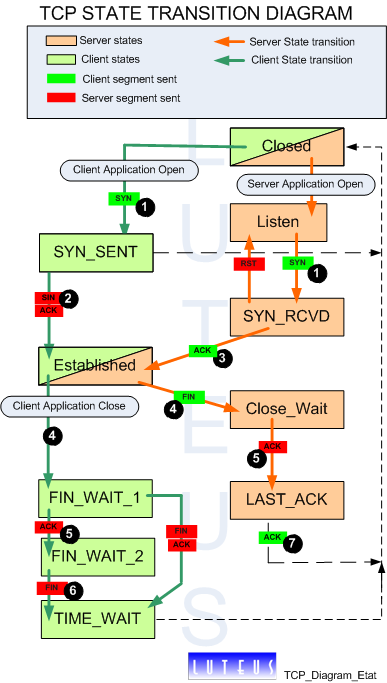
# Tester TCp-IP - netstat

## Netstat:

Donc **netstat** en commande de base, permet de connaître des statistiques sur les protocoles TCP-IP- UDP...



Les valeurs possibles fréquentes sont

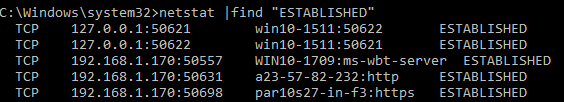
**ESTABLISHED** un socket de connection est établi

**TIME\_WAIT** la connexion est en attente après fermeture pour repasser en statut CLOSE (fermé)

**CLOSE\_WAIT** la connexion distante est tombée, on attend les paquet de fermeture « propre »

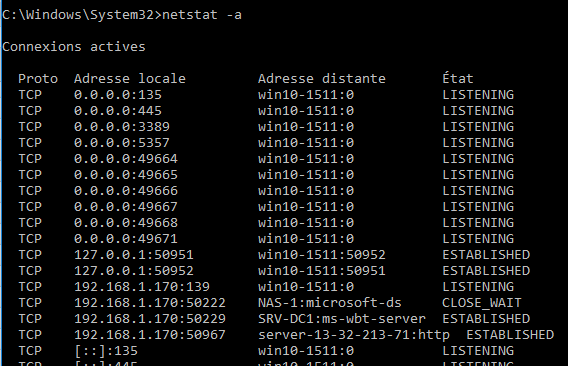
**LISTEN** un socket est etabli, et on attends de recevoir des paquets (il faut demander une option –l ou –a pour voir ces ports en écoute

**N.B** : Du coup on peut filtrer les sorties avec un **|find "mot clé"** comme dans **netstat |find « ESTABLISHED »**

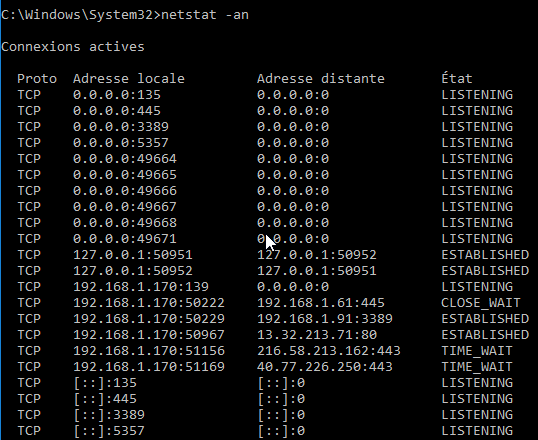


## Netstat -a n port en écoute:

L'option **netstat -a** permet d'ajouter les ports en "ecoute" LISTENING

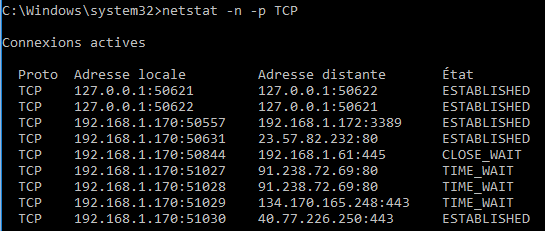


**L'option -n** affiche les n**° de port IP** au lieu de tenter une résolution de nom.



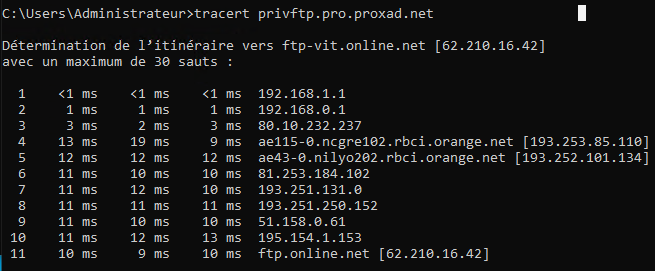
## Netstat -a –p TCP port en écoute par protocole:

**L'option -p** permet de filtrer un protocole (**TCP** ou **UDP**))



## Test liaison ftp – affichage dans Netstat –an :

Sur une machine ayant la possibilité de sortir, pour atteindre un poste en **62.210.16.42** (serveur **FTP** hebergeur de ***ONLINE***). 1 **Tracert** permet de vérifier l'adresse **IP** du serveur **FTP, nommé *privftp.pro.proxad.net***



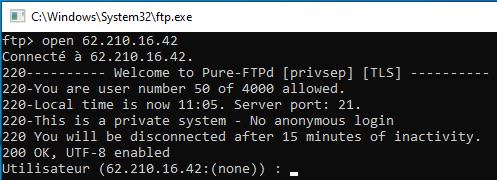
On peut lancer le client **FTP** windows natif

 donnant 

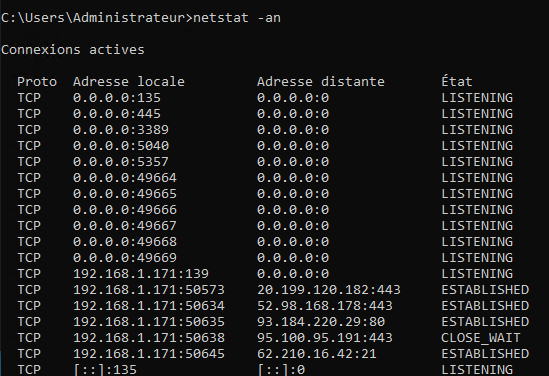
On demande



On est connecté, en attente du mot de passe

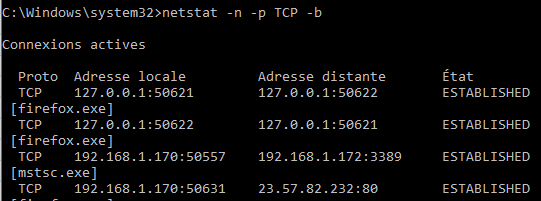


Un **netstat –an** affichera le connexion prise



**Netstat –b executable associé  :**

**L'option -b** permet d'afficher le nom de l'exécutable qui concerne chaque connexion ou port d'écoute.



**Netstat -o PID correspondant  :**

La commande **netstat** permet donc avec les options **–ano** de connaitre les n° de **pid** des processus associés aux n° de **ports**

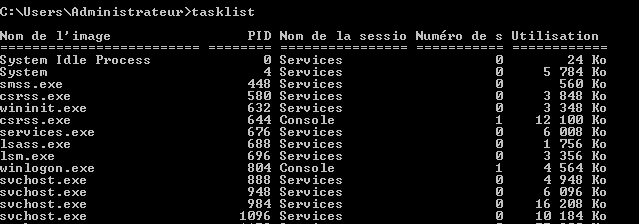


qui utilise le port 668 ?

le PID 1984…

**tasklist /svc et PID:**

Par exemple **netstat –a -o** ou **netstat –a -b** sont très utiles, associées aux utilitaires **tasklists** et **taskill**



le PID 1984…

c'est Carbonite !



et les services sont affichables, avec l'options **/SVC** par exemple ici svchost en PID ***984*** correspondrait à Windows defender… !



On peut ensuite faire le ménage, via

**taskkill /f /im carbonite.exe**

Ou

**taskkill /PID 1984**

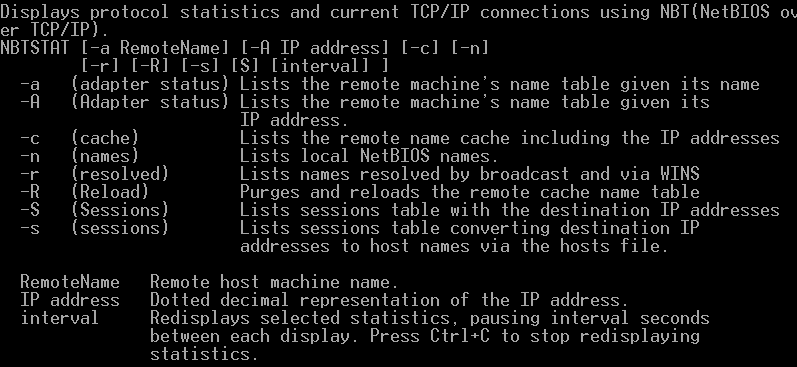
**N.B**: L’argument **/f** force les processus à se terminer

**N.B**: l’argument **/im** spécifie le nom d’image du processus à terminer (dans l’exemple ci-dessus, le nom d’image du processus est « ***carbonite.exe*** »)

## Nbtstat –n :

Les essais sur une configuration peuvent se faire à bas niveau, directement au niveau d’un boite DOS

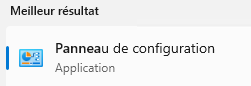
**Permet de connaître des statistiques sur NETBIOS SUR TCP/IP**



# Telnet test de socket

## Installation telnet:

Le protocole **telnet** n’est pas installé par défaut… il faut l’ajouter sur notre poste Windows… **N.B** : Sous Windows 11 le client **Telnet** n’apparait même pas dans la recherche.

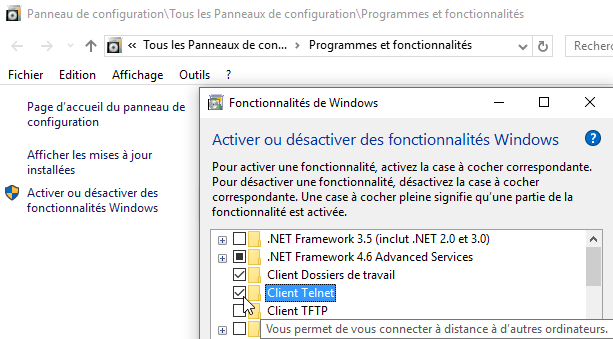


Via l'ancien **panneau configuration**

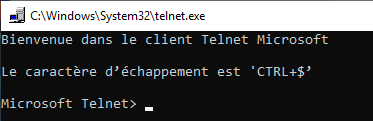
Ou **– control.exe**

On demande **Panneaux de Configuration / Programmes et Fonctionnalités**

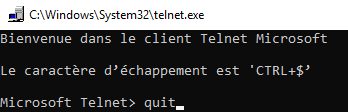
**Activer ou Désactiver des fonctionnalités Windows / Client Telnet**



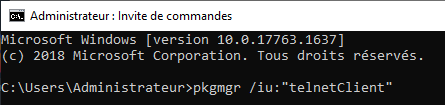
Désormais, l’application **Telnet** sera disponible.

On quittera une session Telnet par la commande **quit**

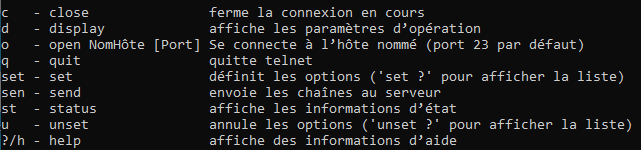


**N.B**: si on veut rendre **telnet** plus accessible, **pkgmgr /iu:"TelnetClient"**



## Telnet - Test de Socket = @IP+ port distant:

L'utilisation de **telnet** est assez basique



On peut tester ensuite des **socket IP** (couple **Adresse Ip** + **N° de Port**) simplement en tapant une commande du genre

### Port 3389 (RDP)

Par exemple pour le port **TCP 3389** qui correspond à du **RDP**, (Bureau à distance) que l’on peut tester sur une machine distante (activation en écoute via les propriétés du poste de travail d’une machine)

**N.B**: sur les anciennes versions windows on tapait directement

**telnet @ip port**



**N.B** : sur les nouvelles versions, on passe en Mode **Telnet** et ensuite on demande **Open** **@ip port**



Si le port 3389 n’est pas ouvert sur la machine 192.168.1.171, on aura alors



Si le port 3389 est en écoute, on aura RIEN… mais… cela veut dire que la connexion est établie ! (on ne peut pas établir une connexion RDP en telnet… Cela peut être une fenêtre noire, peu importe, si la connexion est prise cela veut dire que le port de la machine distante est ouvert….

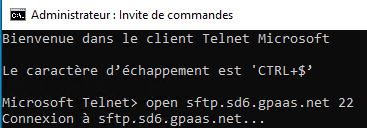
Ou



Ne sera pas visible d'ailleurs si on fait un **netstat -na car la connexion RDP n'est pas montée !**

### Port 22 (SFTP)

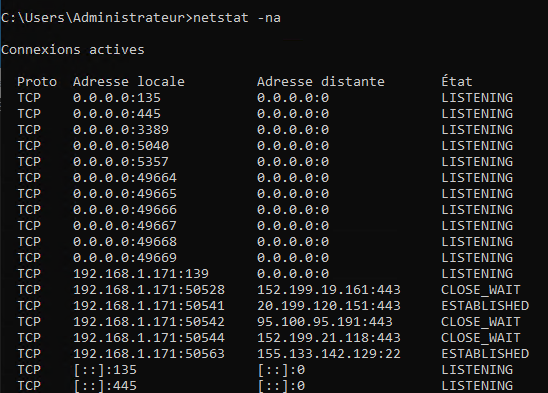
donc si on veut tester **SFTP** sur une machine distante nommée ***sftp.sd6.gpaas.net*** ( adresse ip ***155.133.142.129***) on testera le port **22**



le serveur SFTP répond…

 le port est ouvert

Visible d'ailleurs si on fait un **netstat -na**

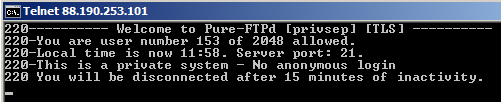


### Port 21 (FTP)

donc si on veut tester FTP sur une machine distante 88.190.253.101 on tentera le port 21



Et si on obtient



On sait que l’on a ouvert une connexion, …

Si le port 21 est fermé, on aura

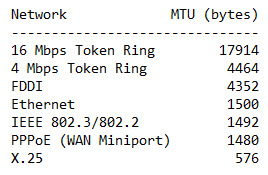


# Tester TCP/IP - Compléments

## Test MTU ping -l -f:

Dans un **ping**, l''option -**l** permet de spécifier la taille des paquets

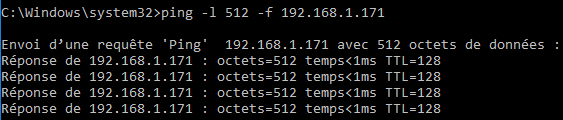
Dans un **ping**, l''option -**f** demande de ne pas fractionner le paquet



**Ping –l taille -f**

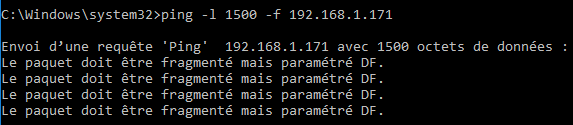
**-l** Permet de spécifier la taille des paquets et **–f** demande de ne pas fractionner le paquet, cela va permettre de tester la taille maximale possible.

A **512 octets**, cela marche, pour tout le monde en général



A **1500 octet** cela ne marche plus… pourquoi ?

Car chaque protocole ajoute des octets par rapport aux octets de data "pur", c'est le principe d'encapsulation. Et donc un paquet de 1500 octets utiles, spécifié par –l 1500 necessite en fait l'envois d'une trame de plus de 1500 octets, qui dépassera du coup la limite maximale.

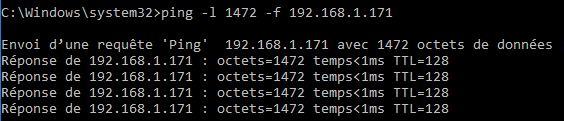


**N.B**: comme on ne s'ait pas par quel routeur on va passer, ni quel protocole on peut prendre à un moment donné, il vaut mieux fixer la taille maximale avec un seuil de sécurité !

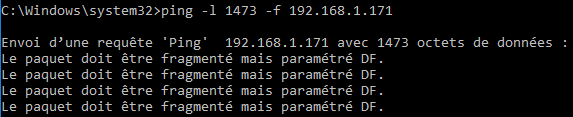
### Constat de la valeur MTU 1500 en Wan

Ainsi une trame **Ethernet** comme la commande **Ping** qui possède 1 en-tête de 28 **octets** (20 pour entête Ip et 8 pour entête ICMP) va avoir une taille maximale transmissible de 1500 – 28 = 1472 **octets.**

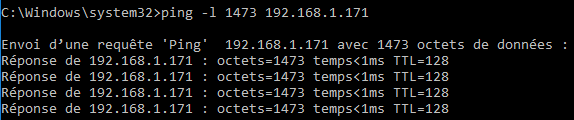
Donc 1472 octets à transmettre via la commande ICMP ping cela marche



Mais 1473 octets cela ne marche plus !

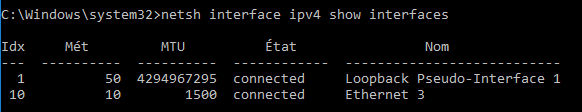


Evidemment sans **/f**, cela marche car le paquet sera fragmenté.

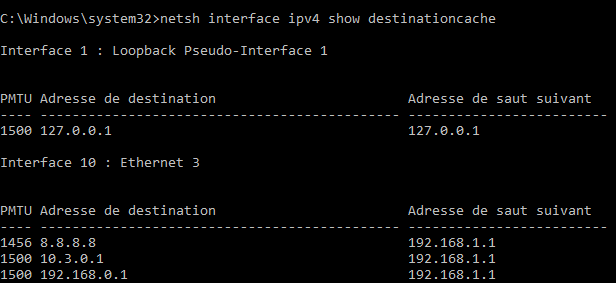


Si on doit transmettre la même quantité d'octet, via **FTP**, (qui n'est pas **ICMP** mais qui est un protocole applicatif, surcouche **IP**) il faut comprendre que en plus de **l'entête IP et TCP**, il faudrait rajouter les informations liées au **protocole TFTP**….). Pour faire simple, dès que l'on prend une connexion physique autre que la liaison **Ethernet 802.3 Lan** (par exemple **ADSL, FIBRE, 4G/LTE**) alors la taille en octets de données réellement émissible sans fragmentation varie et diminue. **TCP-IP** peut modifier dynamiquement au cours d'un échange la taille **MTU** à travers un mécanisme nommé **MSS maximun Segment size.**

Pour information en **LAN : netsh interface ipv4 show interfaces**

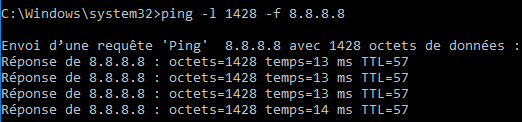


et en **WAN : netsh interface ipv4 show destinationcache**

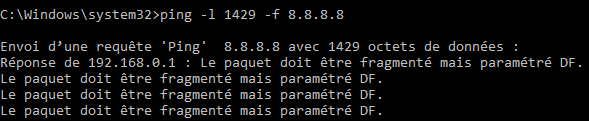


Ainsi une trame **Ethernet 802.3** comme la commande **Ping** qui possède 1 en-tête de 28 **octets** (20 pour entête Ip et 8 pour entête ICMP) va avoir une taille maximale transmissible ici de 1456 – 20 - 8 = 1428 **octets**

Une **MTU** est posée à 1428 pour **Windows** des que l'on "sort"

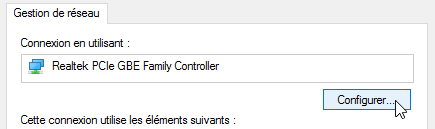


et

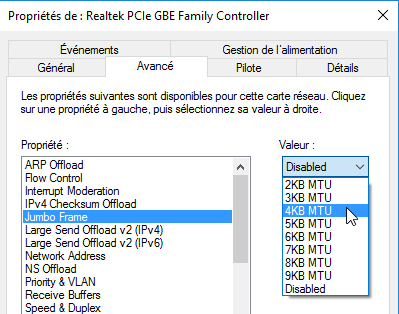


### Jumbo Frames - MTU en Lan

Si on modifie la taille des paquets **MTU** dans les paramètres de la carte réseau

 **Configurer…**

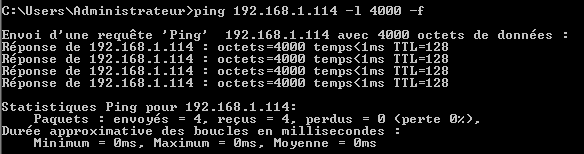
Et on demande des trames dites **Jumbo Frame** maxi **MTU de 4ko**



On devrais pouvoir faire un ping avec un paquet de 4 kilo…

**Ping 192.168.1.114 -l 4000 -f**

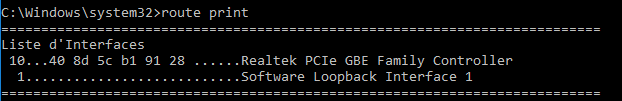
Si cela ne marche pas, c’est que entre les 2 postes un appareil (hub, switch) n’accepte pas ces trames agrandies… on peut essayer directement avec un câble croisé.



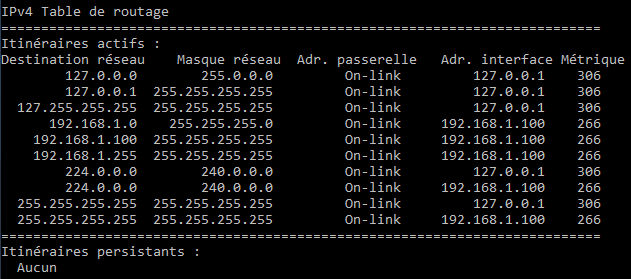
## Routage route print netstat –r :

On est sur une machine en 192.168.1.100 sans indication de

La commande **route print,** ou **netstat -r** donne l'affichage suivant



Et la table par défaut étant



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Destination Réseau**  Vers ou veut on aller | **Masque Réseau**  permet de définir une plage étendue de destination | **Adr. Passerelle**  Adresse /carte par laquelle on doit passer | **Adr. interface**  à partir de quelle adresse / carte on part | **Métrique**  Cout. plus il est petit, plus la route est prioritaire |

**On-link** = aucun routage nécessaire, on est directement relié au réseau

**192.168.1.100 / 255.255.255.255** : Route de l'ordinateur vers lui-même, "Destination réseau" et "Adresse interface" ont la même valeur. le masque entièrement à 255 qui permet de désigner une plage limitée à une seule adresse.

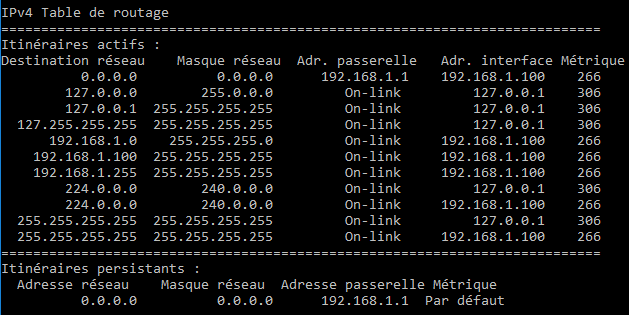
**192.168.1.0 / 255.255.255** permet d'indiquer les adresses du même réseau IP (pas de routage necessaire, forcément)

**192.168.1.255 / 255.255.255.255** permet d'indiquer l'adresse de broadcast

Toutes les autres entrées servent à indiquer les adresses le multidiffusio. (127.0.0.0 = localhost – 127.255.255.225 etc etc )

L'ordre de traitement de la table de routage va des masques les plus longs aux plus petits. C'est à dire que le routeur va d'abord comparer les sous-réseaux avec le masque 255.255.255.255 pour finir par comparer les sous-réseaux avec le masque 0.0.0.0.Il peut y avoir plusieurs routes possibles, mais elles n'ont pas la même métrique

Si on ajoute une passerelle en 192.168.1.1 alors

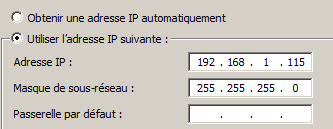
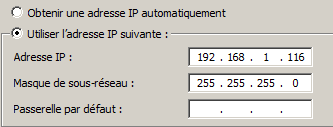


Route par défaut **0.0.0.0/0.0.0.0** : c'est la route utilisée si aucune autre route possible n'a été trouvée dans la table de routage

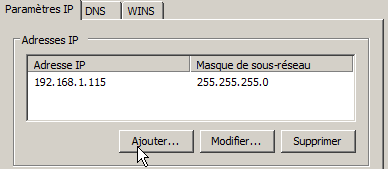
## Test routage route add :

soit 2 machines respectivement communiquant

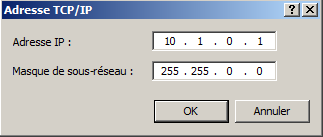
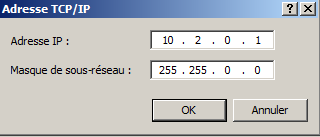
en 192.168.1.115/24 et 192.168.1.116/24

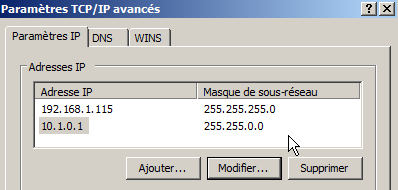
on ajoute sur chaque poste une 2° @ IP respectivement en 10.1.0.1 et 10.2.0.1. donc via **Avancé/ Ajouter**



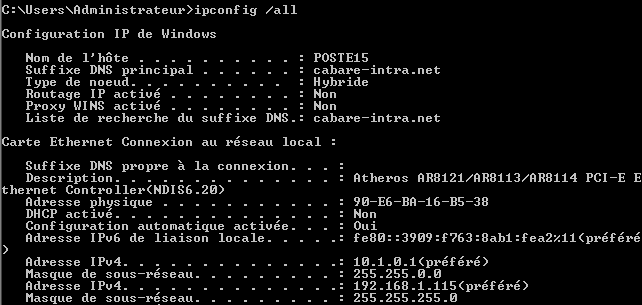
Donc a joute sur la 1° machine et sur la 2° machine

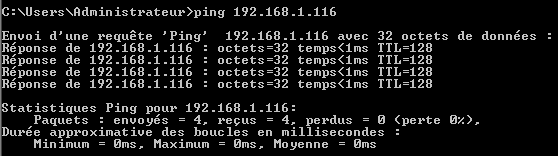
On aura donc pour la 1° machine



Et pour la 1° machine on peut vérifier



**Ping** en 192.168.1.116 marche, bien sur



**Ping** en 10.2.0.1 échoue

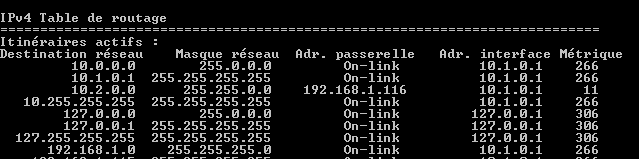


Si on ajoute une route, il faut connaitre l’adresse ip en 192.168.1.116 de la machine d’à coté, que l’on utilisera comme moyen d’accès…

**Route add 10.2.0.0 mask 255.255.0.0 192.168.1.116**



Et on vérifie notre table



Maintenant cela marche

